



ESTUDI D'INTERFÍCIES MULTITÀCTILS I MULTIUSUÀRI

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Antoni Gurguí Valverde
i dirigit per
Enric Martí Gòdia
Bellaterra, 18 de Setembre de 2009



El sotasignat, Enric Martí Gòdia

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Antoni Gurguí Valverde

I per tal que consti firma la present.

Signat:

Bellaterra, 18 de Setembre de 2009

| | |
|--|----|
| 1 Introducció | 1 |
| 1.1 Què són les interfícies multitàctils? | 2 |
| 1.2 Dispositius tàctils i multitàctils existents | 3 |
| 1.3 Motivació i Objectius | 16 |
| 2 Funcionament de la taula | 18 |
| 2.1 Processament de la imatge..... | 19 |
| 2.1.1 Filtre de color a blanc i negre | 20 |
| 2.1.2 Filtre d'extracció del fons | 20 |
| 2.1.3 Filtre de suavitzat | 21 |
| 2.1.4 Filtre passa altes..... | 21 |
| 2.2 Extracció i seguiment de les taques..... | 22 |
| 2.3 Transmissió de les dades | 24 |
| 3 Construcció i implementació..... | 26 |
| 3.1 Construcció d'una taula multitàctil basada en visió per computador | 26 |
| 3.1.1 Consideracions de disseny | 27 |
| 3.1.2 - Mètodes d'il·luminació per a taules multitàctils basades en visió per computador..... | 30 |
| 3.1.3 Construcció d'una taula multitàctil basada en visió per computador..... | 33 |
| 3.2 - Implementació..... | 42 |
| 4 Resultats | 48 |
| 5 Conclusions i millores | 51 |

1 Introducció

Un aspecte clau en l'ús dels ordinadors és la forma d'interacció entre l'usuari i la màquina, és a dir, la manera en què l'ordinador i la persona es comuniquen. Inicialment es basava en l'entrada de text per teclat que l'ordinador interpretava. Posteriorment, l'aparició de les interfícies gràfiques basades en finestres, juntament amb perifèrics com el ratolí, van facilitar aquesta interacció, fent que el nombre d'aplicacions possibles i el ventall de persones capaces d'utilitzar-les augmentés enormement. Més recentment, han aparegut dispositius com les PDA's i TabletPC's que obren les portes a l'escriptura a mà digital o a la selecció mitjançant un dit, o com l'iPhone, que ens permet interactuar-hi amb més d'un dit.

Així, actualment, estem veient un gran interès, tant per part dels usuaris com per part de les empreses, per explorar noves interfícies i noves maneres d'interactuar amb els dispositius electrònics. Un exemple són les interfícies multitàctils. La idea bàsica d'aquestes és la de permetre a l'usuari utilitzar directament les mans per interactuar amb el sistema i així manipular les dades. És més, aquest tipus d'interfícies poden ser utilitzades per més d'un usuari simultàniament. D'aquesta manera, es poden crear murs interactius en els què hi pot actuar tota una classe de primària, instruments musicals col·laboratius o entorns d'usuari per a persones discapacitades i un llarg etcètera encara per descobrir.

En aquest capítol es presenta:

- **Què són les interfícies multitàctils** i què aporten respecte les tàctils.
- **Dispositius tàctils i multitàctils existents**, on es repassa breument la història d'aquestes interfícies.
- **Motivació i objectius** del projecte.

1.1 Què són les interfícies multitàctils?

Les interfícies multitàctils són aquelles que permeten a l'usuari interactuar directament amb allò que veu utilitzant més d'un dit simultàniament. És a dir, l'usuari pot tocar la pantalla en diversos punts i obtenir una resposta a cada un dels contactes. Això implica que l'usuari no utilitza cap dispositiu intermig: ni ratolí, ni teclat. A més, aquestes interfícies permeten que més d'un usuari interaccioni al mateix temps.

Les interfícies tàctils, per altra banda, són aquelles que permeten interactuar només amb un dit o un punter simultàniament. Aquestes ja fa un cert temps que s'utilitzen, i han demostrat ser força acceptades pel públic en general (ja que permeten una interacció més directa). Les trobem sobretot en dispositius on el ratolí i el teclat són incòmodes d'utilitzar (dispositius mòbils, caixes enregistradores de bars i restaurants, etc.).

Així, les principals diferències entre les interfícies tàctils i les multitàctils és el número de dits que pot utilitzar l'usuari simultàniament i la possibilitat d'interactuar més d'un usuari al mateix temps. Per tant, aquesta interacció permet:

- Seleccionar la funció a executar a partir del numero de dits utilitzats (i no de la opció d'un menú),
- Poder seleccionar múltiples punts al mateix temps,
- Distingir dos contactes diferents quasi simultànies.

Així, actualment existeixen diferents tècniques per detectar el contacte dels dits en una pantalla. En aquest treball s'estudia i utilitza la tècnica basada en visió per computador. El dispositius multitàctils basats en aquesta tècnica comparteixen les següents idees:

- Utilitzar una càmera per detectar el contacte,
- Detectar el contacte utilitzant la reflexió de la llum (ja sigui visible o infraroja) ,
- La superfície de contacte i la de projecció són la mateixa (no hi ha tecnologies ni capes pel mig),
- Utilitzar la capacitat dels ordinadors per processar imatges i extreure'n les característiques desitjades en temps real.

Per altra banda, aquest mètode permet utilitzar objectes amb marques fiduciàries (de l'anglès *fiducial marks*). Les marques fiduciàries són dibuixos geomètrics que permeten a un sistema de visió detectar-les, distingir-les i saber-ne la seva posició i rotació (figura 1.11b). Aquestes marques s'utilitzen en diferents camps, però en les interfícies multitàctils basades en visió per computador serveixen per crear controls físics (en contraposició amb els virtuals creats per l'aplicació). Podem trobar un bon exemple de l'ús d'aquestes marques en el *racTable* [JGA05] i en entorn de realitat augmentada [KaB99].

1.2 Dispositius tàctils i multitàctils existents

La tecnologia multitàctil no és una tecnologia que acabi d'aparèixer. Deixant de banda el teclat, que permet prémer més d'una tecla al mateix temps, es pot dir que des dels anys 60 s'han anat creant interfícies i tecnologies tàctils i multitàctils. A continuació es fa un breu repàs de les més importants [Bux09].

El 1972 es va presentar el *PLATO IV Touch Screen Terminal* [She05]. Aquest era un ordinador utilitzat en el món de l'educació. Incloïa un panell tàctil, que consistia en una malla IR de 16x16 punts que permetia a l'usuari seleccionar una part de la pantalla (fig. 1.1). D'aquesta manera es podien passar tests als alumnes i aquests responien tocant la resposta correcta. Tot i que no permetia l'ús de més d'un dit a la vegada, es considera la primera aplicació en informàtica de la tecnologia tàctil.



Figura 1.1: Exemple d'ús del PLATO IV Touch Screen Terminal

El 1983, M. Krueger va presentar el *Video Desk* [KGH85]. Aquest utilitzava una càmera de vídeo i processament d'imatge per seguir la posició de la mà en 2D i detectar característiques de la imatge com la mà, un dit o l'orientació d'alguns d'aquests. Encara que no és estrictament tàctil, la idea del seguiment usant visió per computador ha estat molt utilitzada (fig. 1.2). A més, l'autor ha treballat molt en temes de gestos i interacció amb més d'una mà i multiusuari.

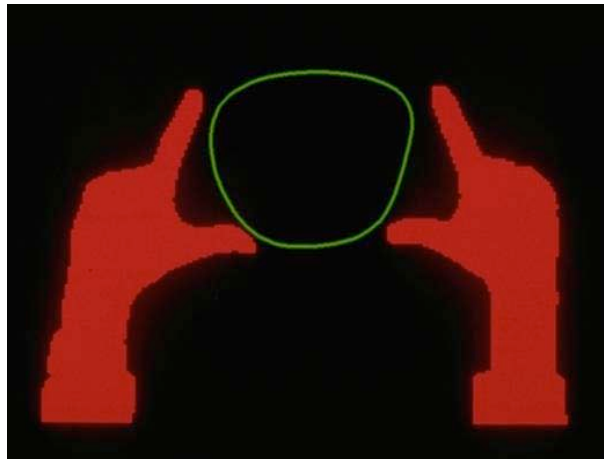


Figura 1.2: Sortida del Video Desk

Aquest mateix any, Hewlett-Packard va presentar l'HP-150 [HP150], un ordinador personal que portava una pantalla de 9 polzades de Sony envoltada per emissors i receptors d'infrarojos, de manera que podia detectar la posició del dit (o qualsevol objecte opac) en contacte amb la pantalla (fig. 1.3).



(a)



(b)

Figura 1.3: a) HP-150 i b) Exemple d'ús d'aquest

El *Multitouch-Tablet* [LBS85] va ser desenvolupat el 1985 per l'Input Research Group de la Universitat de Toronto. Utilitzava capacitància per detectar el tacte (veure figura 1.4a). Era capaç de detectar qualsevol nombre de punts de contacte simultàniament. L'any següent, en W. Buxton i en B. Myers, de la mateixa universitat, van presentar els resultats d'un estudi on es demostrava els beneficis de l'ús de dues mans en tasques de CAD (fig. 1.4b) [BuM86].



Figura 1.4: a) Multitouch-Tablet. b) Subjecte en un dels experiement de l'estudi de Buxton, W. i Myers, B.

El mateix any es va crear el *Sensor Frame* [Mca86] a la Carneige Mellon University. Utilitzava sensors òptics a les cantonades del marc per detectar els dits, tal i com es veu a la figura 1.5. Podia detectar fins a 3 dits simultàniament, encara que, a causa de la tècnica utilitzada per a la detecció, era propens a equivocar-se perquè les ombres el feien confondre.



Figura 1.5: Exemples d'ús del Sensor Frame.

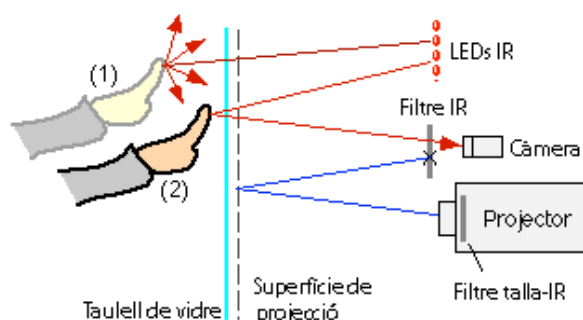
La *Digital Desk* va ser presentada l'any 1991 [Wei91]. Aquesta és un dels primer intents de crear un sistema amb projecció des de dalt. Utilitzava tecnologies òptiques i acústiques per captar el contacte dels dits i les mans, així com objectes, com per exemple, controls fets de paper. Va demostrar conceptes multitàctils com l'ús de dos dits per escalar i fer traslacions.

El 1997, Matsushita et al. van presentar la *HoloWall* [MaR97]. Aquesta és una paret o plafó sensible als dits, a la mà i a qualsevol objecte. Aquesta paret està composta per un vidre amb un material per a la projecció al darrere, on també hi ha un projector que s'utilitza per mostrar la imatge. En aquest mateix costat es troba una font de llum IR i una

càmera sensible a aquest espectre. Quan l'usuari toca la paret, el dit reflexa la llum IR que és capturada per la càmera, tal i com es pot observar a la figura 1.6b. A més, permet interactuar a més d'un usuari simultàniament (fig. 1.6a).



(a)



(b)

Figura 1.6: a) Exemple d'ús de l'HoloWall. b) Esquema de funcionament.

L'empresa dedicada a taules digitalitzadores *Wacom*, l'any 1998, va presentar el model *Intuos 1*. Tres anys més tard va presentar la *Intuos 2* (Fig1.7). Aquestes permetien controlar fins a 10 graus de llibertat amb les dues mans: les coordenades x i y del llapis, la inclinació del llapis en l'eix x i y , la pressió de la punta, el valor d'un dial muntat en l'aerògraf, la posició i rotació del ratolí cursor (*puck*, en anglès) així com la rotació d'una rodeta muntada en aquest. Actualment aquesta empresa és una de les principals desenvolupadores de taules digitalitzadores (tant independents com muntades en una pantalla de *tabletpc*).



(a)



(b)

Figura 1.7: a) Taula Wacom Intuos 2 amb el llapis i el ratolí cursor. b) Aerògraf.

L'any 2001 el Mitsubishi Research Labs va crear la taula *DiamondTouch* [DiL01]. Aquest és un dispositiu d'entrada sensitiu multitàctil i multiusuari. La taula funciona enviant senyals elèctrics a la matriu de files i columnes inserides a la superfície. La

DiamondTouch permet que interactuïn fins a 4 persones simultàniament. Cada usuari s'asseu en una cadira que disposa d'un receptor. L'usuari toca físicament la taula. Quan fa contacte físic, un petit corrent elèctric passa des de la matriu fins al receptor de la cadira, a través del seu cos (veure la figura 1.8a). Així el mòdul de recepció pot determinar quines zones són tocadetes per cada usuari. La *DiamondTouch* utilitza un projector per projectar la imatge sobre la taula, tal i com es veu a la figura 1.8b. El sistema permet als usuaris controlar les aplicacions amb gestos. És capaç de detectar gestos per contacte i gestos basats en la forma de la mà.

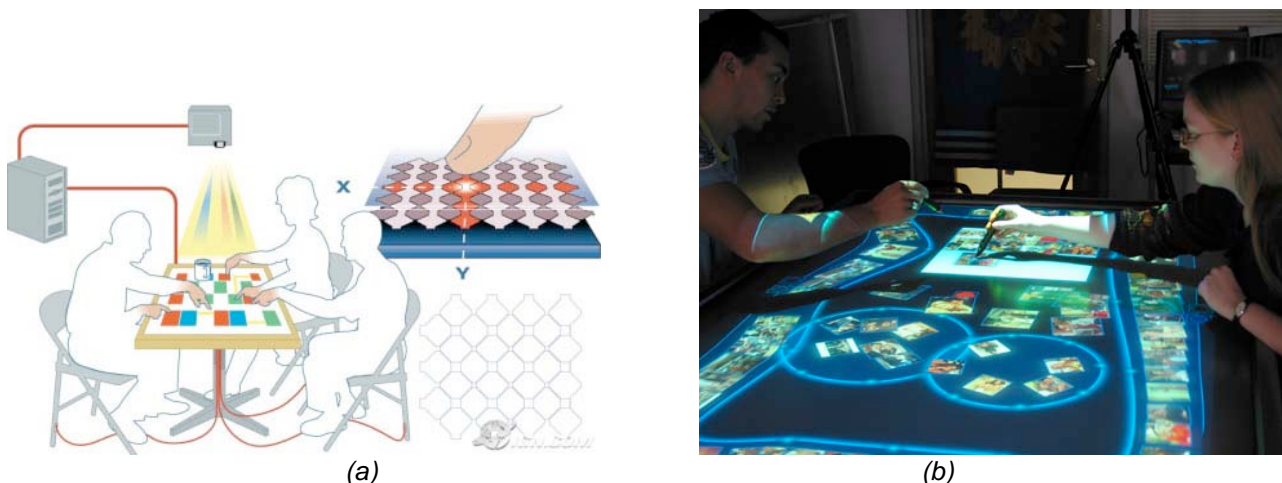


Figura 1.8: a) Esquema del funcionament de la DiamonTouch. b) Exemple d'ús.

La taula *SmartSkin* (fig. 1.9) és un dispositiu d'entrada sensitiv multitàctil presentat per Jun Rekimoto l'any 2002 [Rek02]. La capa sensitiva consisteix en un emissor en forma de reixeta i d'electrodes sensors (cables de coure). Els cables verticals són electrodes emissors i els horitzontals receptors. Cada cert temps s'emet un senyal a través d'un emissor (només un per interval). El receptor és capaç de rebre el senyal ja que actua com un condensador. Quan l'usuari toca amb el dit la malla, provoca una atenuació del senyal. Mesurant aquests canvis es detecta el contacte. La imatge és projectada damunt de la taula i, si la xarxa és molt densa, el sistema és capaç de distingir objectes. Encara que no utilitza imatges captades per una càmera, aquest sistema utilitza mètodes de processament d'imatge com anàlisi de regions connectades o interpolació bicúbica per tal d'obtenir la forma d'allò que ha fet contacte (la mà, els dits o qualsevol altre objecte).

La *Touchlight* (fig. 1.10) és un dispositiu d'entrada basat en una càmera que permet la interacció basada en gestos [Wil04]. Aquest dispositiu va ser desenvolupat per Andrew D. Wilson, del Microsoft Research, a finals de l'any 2004. Utilitza una superfície feta amb acrílic a la que s'enganxa un material per a la retroprojecció. Darrere de la superfície es col·loca un projector. Per eliminar el fons, el sistema utilitza dues càmeres i una font

d'infrarojos per detectar la interacció. Quan l'usuari acosta la mà a la superfície, aquesta reflexa la llum infraroja. Les càmeres estan col·locades en diferents angles, una a la dreta i l'altra a l'esquerra de la superfície de contacte i projecció. A causa de les seves posicions, s'han de corregir les distorsions de lents i de perspectiva de les imatges obtingudes. Quan s'uneixen les dues imatges en una, el objectes propers a la pantalla coincidiran i quedaran ressaltats, i els objectes més allunyats no coincidiran i es veuran menys. A diferència de les interfícies anteriors, aquest sistema funciona més com a dispositiu basat en gestos que no pas com a dispositiu multipunt. A més, aquest dispositiu disposa d'un micròfon per detectar vibracions en la pantalla. Això permet als usuaris interactuar picant a la pantalla, a mode de clic de ratolí.

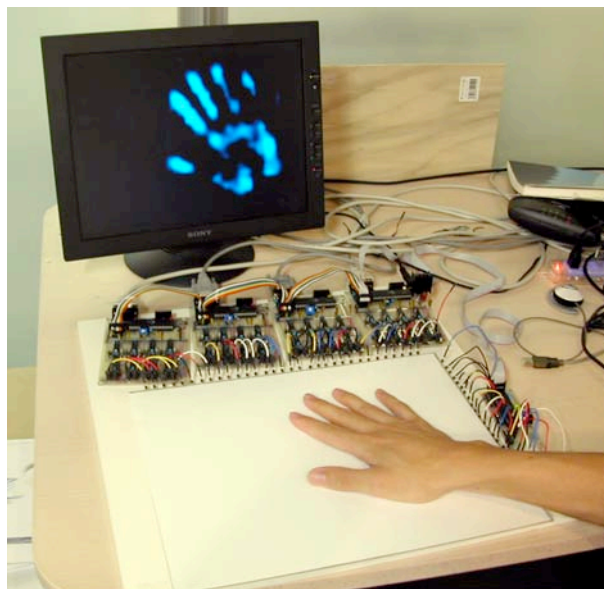


Figura 1.9: Exemple de funcionament de l'SmartSkin



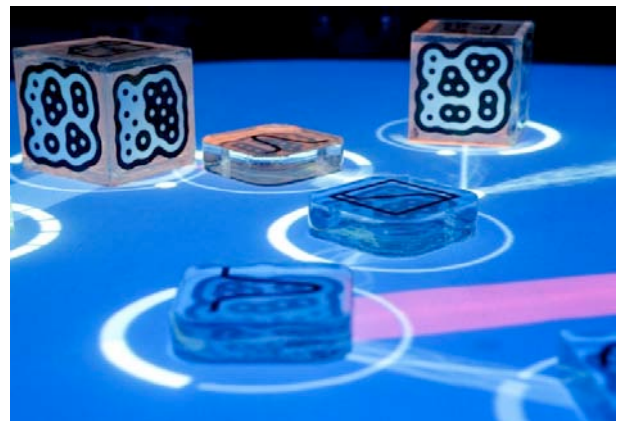
Figura 1.10: Exemple d'ús del Touchlight

L'any 2005 es va presentar el *reactTable* [JGA05]. El *reactTable* és un instrument musical electrònic col·laboratiu amb una interfície multitàctil tangible. La taula és capaç de seguir marques fiduciàries que permet als usuaris afegir instruments i controlar-los mitjançant la rotació i posició d'aquestes (fig. 1.11). Movent i combinant múltiples marques

es poden crear sons i ritmes. La taula utilitza emissors de llum infraroja per il·luminar la superfície. Quan un usuari afegeix un objecte amb una marca damunt de la taula, aquest reflexa el patró a la càmera que serà reconegut i processat per l'ordinador. L'última versió del *reactTable* permet interacció multitàctil amb els dits.



(a)



(b)

Figura 1.11: a) *reactTable* b) Exemple de marques fiduciàries del *reactTable*

La *JazzMutant Lemur* és un dispositiu d'interacció multitàctil, extern a l'ordinador, dissenyat per controlar aplicacions d'àudio i multimèdia [Lem] (fig. 1.12). Com que tots els controls (rotatius i lineals) són creats virtualment a la pantalla, el dispositiu permet a l'usuari personalitzar la interfície segons les seves necessitats. La *Lemur* es comunica a través d'una interfície *ethernet* amb l'ordinador que executa l'aplicació amb la que l'usuari treballa.

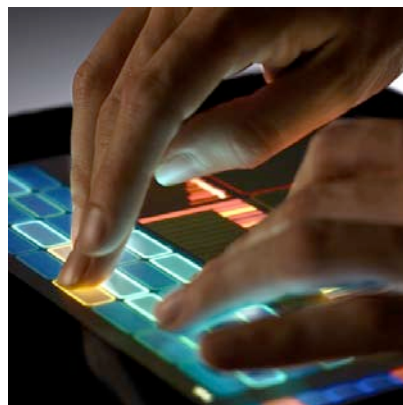
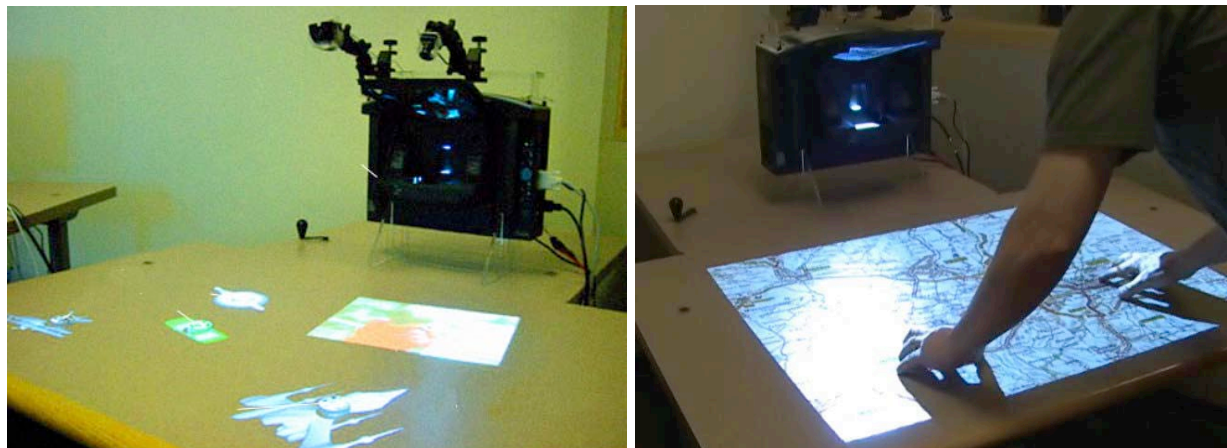


Figura 1.12: Exemple d'ús de la *Lemur*

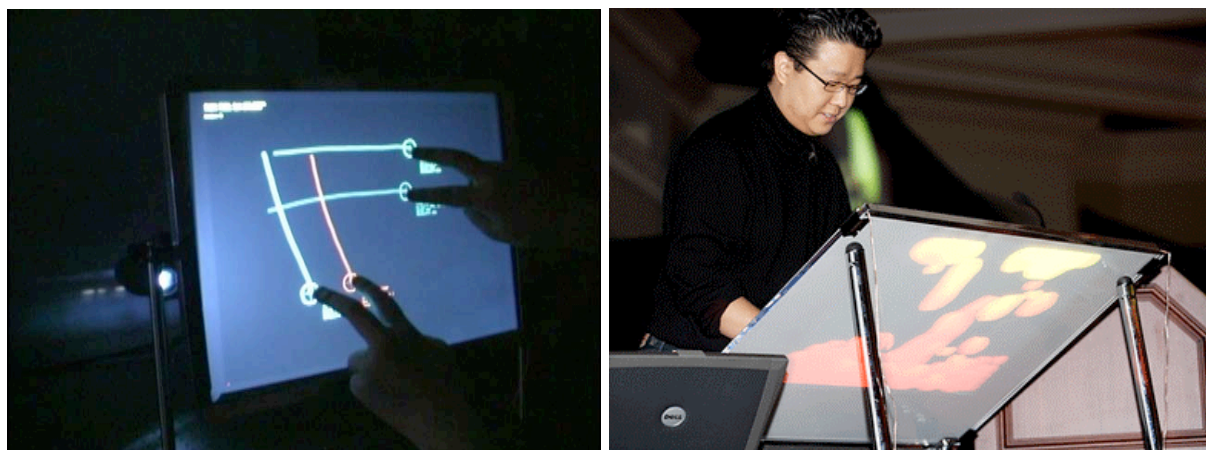
Un altre dispositiu creat per Wilson, l'any 2005, és el *PlayAnywhere* [Wil05]. El *PlayAnywhere* és un dispositiu d'interacció multitàctil basat en una càmera amb projecció

des de dalt, tal i com es veu a la figura 1.13. Aquest sistema utilitza un projector per presentar l'entorn o aplicació i una càmera de vídeo per veure la taula des de dalt, il·luminada amb IR. Quan l'usuari interactua amb la taula, les mans queden il·luminades. Per poder seguir els dits, se n'analitza el contorn. Per detectar el contacte es compara el contorn de la punta del dit i el de la seva ombra. Si es detecta l'ombra a prop de la punta del dit, el sistema ho interpreta com una passada per damunt sense tocar (de l'anglès *hover*). Si no hi ha ombra, el sistema determinarà que s'està fent contacte amb els dits. El principal avantatge d'aquest sistema és que es pot utilitzar sobre qualsevol superfície.



(a) (b)
Figura 1.13: a) PlayAnyWhere . b) Exemple d'ús.

El mateix any en Jeff Han va presentar una taula multitàctil de baix cost basada en l'ús d'una càmera [Han05]. El sistema utilitza la tècnica anomenada **Reflexió Interna Total Frustrada** (de l'anglès FTIR, *Frustrated Total Internal Reflection*) que consisteix a capturar la llum infraroja dins d'una làmina acrílica. Quan es toca la làmina, la zona tocada canvia el patró de reflexió de la llum, provocant que aquesta sigui difosa en comptes de reflectida. Si apuntem una càmera a la làmina és possible detectar la llum infraroja que surt i reconèixer un contacte. L'any 2006 va crear la seva pròpia empresa, PerceptilePixel.



(a) (b)
Figura 1.14: a) Exemple d'ús . b) Jeff Han durant una presentació de la tècnica

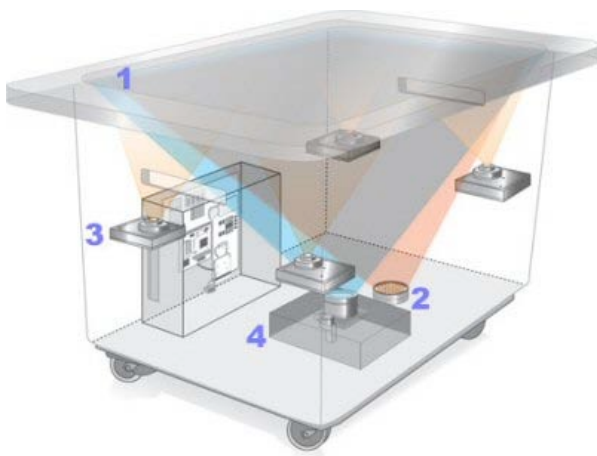
A finals de l'any 2006 es va formar el Natural User Interface Group [Nui06]. Aquesta és una comunitat dedicada a experimentar i desenvolupar noves idees en el camp de les interfícies multitàctils. Des de llavors, la comunitat ha crescut molt i han desenvolupat força codi obert, com per exemple, el *traker* utilitzat en aquest projecte.

A principis del 2007, Apple va presentar un nou telèfon mòbil: l'Apple *iPhone* [AiPn]. Aquest va ser el primer mòbil a permetre una interacció multitàctil, ja que la resta de mòbils tàctils del moment només la permetien fer amb un sol dit. L'*iPhone* detecta el contacte utilitzant camps elèctrics. Quan es fa contacte, el dispositiu detecta el canvi en el camp elèctric que això provoca. Això permet a l'*iPhone* detectar quina part de la pantalla està sent utilitzada. A partir d'aquest dispositiu, Apple ha fet una versió multitàctil del seu reproductor multimèdia *iPod*: l'Apple *iPod Touch* [AiPd].



(a) (b)
Figura 1.15: a) iPhone. B) Exemple d'ús (*Pocket Guitar*).

A mitjan el mateix any, Microsoft va presentar la seva versió de taula multitàctil: l'*MS Surface* [Sur07]. Aquesta taula té l'aspecte d'una tauleta de sofà amb una superfície interactiva, com es veu a la figura 1.16b). La tecnologia utilitzada és similar a la de la *HolloWall*. Un difusor és enganxat al material de la pantalla. La taula s'il·lumina amb llum infraroja des de darrere. Quan un usuari toca la taula reflexa la llum infraroja que és captada per les càmeres de dins de la taula. Com que utilitza més d'una càmera, la resolució d'entrada és prou gran com per poder detectar objectes. A més, Microsoft ha demostrat com es poden utilitzar marques fiduciaris per identificar objectes i permetre interacció específiques.



(a)
Figura 1.16: a) Esquema de funcionament de l'MS Surface. b) exemple d'ús.

Uns mesos més tard, després de la presentació als medis de l'MS Surface, el Microsoft Research Cambridge va mostrar l'ús de tecnologia multitàctil sobre la pantalla LCD d'un notebook: la *ThinSight* [HIB07] (fig. 1.17). Aquesta està formada per una matriu d'optosensors retro-reflectius col·locada darrere de la pantalla LCD. Cada sensor conté un emissor de llum IR i un detector d'IR. Quan un objecte toca la pantalla LCD reflexa la llum cap al detector. Com que aquesta tècnica es basa en l'ús de llum IR, permet l'ús de marques fiduciàries. A més, permet l'ús de dispositius com pistoles IR o comandaments a distància.



Figura 1.17: Exemple d'ús de la tecnologia ThinSight.

Cap a final de l'any 2007, Johnny Chung Lee va publicar a internet un programa que permetia utilitzar el comandament de la Nintendo *Wii* [Wii] per detectar un llapis amb punta emissora d'IR [Lee07]. D'aquesta manera es pot emular una taula digitalitzadora amb un cost molt baix. A més, el comandament de la *Wii* pot detectar fins a 4 punts, de manera que poden interactuar fins a 4 llapis. Mesos més tard va presentar un programa per utilitzar el comandament per detectar fins a 4 dits a l'aire. Aquests dos programes no permeten distingir el contacte: simplement detecten la presència del llapis o el dit davant del sensor.

A mitjan any 2008, *Dell* presenta la nova línia de *TabletPC*: *Dell Latitude XT*. Aquesta línia porta suport per interactuar a través de l'*stylus* o amb la mà (fig. 1.18). Això ho fa utilitzant la tecnologia *DuoSense* de *N-Trig* [PeM02]. Aquesta és una matriu de cables que detecten on s'ha fet contacte amb el dit. En el cas de l'*stylus*, aquest porta una bobina interna que genera un camp magnètic a l'acostar-se a la malla de la pantalla, que detecta la posició del camp.



(a) (b)
Figura 1.18: a) *Dell Latitude XT* amb el seu llapis. b) Exemple d'ús de la pantalla tàtil.

A finals del mateix any, *Apple* va introduir la capacitat de reconèixer més d'un dit al mateix temps en el *pad* dels seus *MacBook* i *MacBook Pro*, com es pot veure a la figura 1.19.



Figura 1.19: Exemple d'ús del pad dels *Apple MacBook* i *MacBook Pro*.

El mateix any 2008, Seth Sandler va presentar en el seu blog un disseny de taula mutitàtil basada en visió per computador molt senzilla de construir: la *MTMini* [San08]. Aquesta utilitza el mètode d'il·luminació de llum difusa (veure apartat 3.1.2) i els materials necessaris per contruir-la es poden trobar a la majoria de llars (fig 1.20).



Figura 1.20: Exemple d'ús d'una taula MTMini

Seguidament, es presenta una taula a mode de resum i comparació. En ella es comparen les següents característiques dels dispositius anomenats anteriorment:

- L'any de creació,
- La tecnologia utilitzada per detectar el contacte dels dits,
- On està col·locat el sensor (o sensors) que detecten el contacte,
- Si el dispositiu es tàctil o multitàctil, i el número màxim de dits que es poden utilitzar simultàniament,
- Si permet l'ús de marques fiduciaris.

| Nom del dispositiu | Any | Tecnologia de detecció | Posició del(s) sensor(s) | Tàctil/Multitàctil (num. de punts) | Permet l'ús de fiduciaris? |
|--------------------------------|-------------|-----------------------------------|---|---|----------------------------|
| PLATO IV Touch Screen Terminal | 1972 | Emissors i sensors d'IR | Al voltant de la pantalla | Tàctil | No |
| Video Desk | 1983 | Càmera | Damunt de la pantalla | Multitàctil | No |
| HP-150 | 1983 | Emissors i sensors d'IR | Al voltant de la pantalla | Tàctil | No |
| Multitouch-Tablet | 1985 | Capacitància | Darrere de la superfície | Multitàctil (il·limitats) | No |
| Sensor Frame | 1985 | Sensors òptics | Al voltant de la pantalla | Multitàctil (3) | No |
| Digital Desk | 1991 | Tecnologies òptiques i acústiques | Damunt de la pantalla | Multitouch (2) | No |
| HoloWall | 1997 | Càmera | Darrere de la pantalla | Multitàctil | Si |
| Intuos 1 i 2 | 1998 i 2001 | N/A | Darrera de la superfície | Tàctil | No |
| DiamondTouch | 2001 | Capacitància | A la taula i a les cadires dels usuaris | Multitàctil (40 = 4 usuaris simultàniament) | No |
| SmartSkin | 2002 | Capacitància | A la pantalla | Multitàctil (il·limitats) | No |

| | | | | | |
|-------------------------------|------|--|------------------------|--------------------------|-----|
| Touchlight | 2004 | Càmera | Darrere de la pantalla | N/A | N/A |
| reacTable | 2005 | Càmera | Darrere de la pantalla | Multitàtil (il·limitats) | Si |
| Lemur | N/A | N/A | N/A | Multitàtil | No |
| PlayAnywhere | 2005 | Càmera | Darrere de la pantalla | N/A | |
| J. Han (FTIR) | 2005 | Càmera | Darrere de la pantalla | N/A | No |
| iPhone | 2007 | Capacitància | N/A | Multitàtil | No |
| Surface | 2007 | Càmera | Darrere de la pantalla | Multitàtil (52) | N/A |
| ThinSight | 2007 | Emissors i sensors d'IR | Darrere de la pantalla | N/A | No |
| J.C. Lee (Wii) | 2007 | Emissors i sensor d'IR | Davant de la pantalla | 4 | No |
| DuoSense | 2008 | Capacitància per la mà i electroestàtic pel llapis | N/A | N/A | No |
| Pad del MacBook i MacBook Pro | 2008 | N/A | En el <i>touchpad</i> | 4 | No |
| MTMini | 2008 | Càmera | Darrere la pantalla | Multitouch | No |

Taula 1: Dispositius tàctils i multitàctils al llarg de la història.

1.3 Motivació i Objectius

Actualment estem vivint una convergència tecnològica on certs dispositius electrònics obtenen funcionalitats que fins ara no tenien (per exemple neveres amb accés a Internet) i on d'altres apareixen com a resultat d'una fusió entre dos o més preexistents (com els telèfons mòbils amb càmera de vídeo i GPS). En aquesta convergència, s'està imposant d'utilització d'interfícies tàctils o multitàctils, ja que aquestes permeten una interacció més amigable (tant gràficament com a nivell d'ocultació del funcionament intern) i no necessiten més que una pantalla amb capacitat de detectar el contacte dels dits (eliminant la necessitat del ratolí o el teclat físic).

A nivell personal, trobo interessant investigar noves maneres d'interactuar amb els ordinadors. La interacció multitàtil ha estat força estudiada al llarg de la història, però ara com ara no existeixen estàndards de facto com passa amb les GUIs actuals. Això ofereix la possibilitat de obrir nous camins i desenvolupar noves idees.

Per tot això, el principal objectiu d'aquest projecte és l'estudi i la construcció d'una taula multitàtil basada en visió per computador, basada en el reacTable [JGA05] i en el disseny de Jefferson Han [Han05], que utilitza la tècnica d'il·luminació RITF (veure 3.1.2). L'estudi contemplarà tant el programari com el maquinari. La taula ha de permetre interactuar amb una aplicació real, de manera que pugui servir com a plataforma de desenvolupament i presentació d'aplicacions multitàctils.

Per tal de validar el bon funcionament de la taula construïda i veure una aplicació real de la interacció multitàctil, s'adaptarà l'entorn 3D de l'assignatura de Gràfics per Computador 2 de l'ETSE de la UAB [JGM08]. Per fer-ho, s'utilitzaran les llibreries del **reacTable** [KBB05] per a la part del client i el *tracker* **Comunity Core Vision** de NUIGroup [CCV].

Així, els passos a realitzar en el projecte són:

- Estudi del funcionament d'un *tracker* de dits.
- Estudi de les necessitats de construcció d'una taula multitàctil basada en visió per computador.
- Construcció de la taula multitàctil basada en visió per computador.
- Estudi de les llibreries existent per fer aplicacions multitàctils i per adaptar-ne de ja existents.
- Adaptació del l'entorn 3D de l'assignatura Gràfics per Computador 2.
- Programació d'una aplicació per fer la presentació del projecte.

2 Funcionament de la taula

En aquest apartat, i als seus subapartats, s'exposa el funcionament bàsic del programari que fa possible la interacció multitàctil. Aquest programari és independent de l'aplicació amb la que l'usuari treballa. Això és així per independitzar l'aplicació de l'usuari del mètode o sistema amb el que es detecta el contacte dels dits.

Actualment existeixen uns quants programes que implementen el processament de les imatges i el seguiment de les taques [CCV][OpT][ToL]. Encara que cadascun d'ells utilitza uns filtres i una configuració d'aquests pròpia, en aquest apartat es presenta un model general per a la comprensió del procés. Així, encara que no s'ha fet la implementació d'aquest programari, s'ha considerat important incloure una explicació del funcionament bàsic.

Així, el programari que es descriu és l'encarregat del processament d'imatges, del seguiment de les taques i de l'enviament de les posicions d'aquestes a l'aplicació que genera la sortida i la que l'usuari veu.

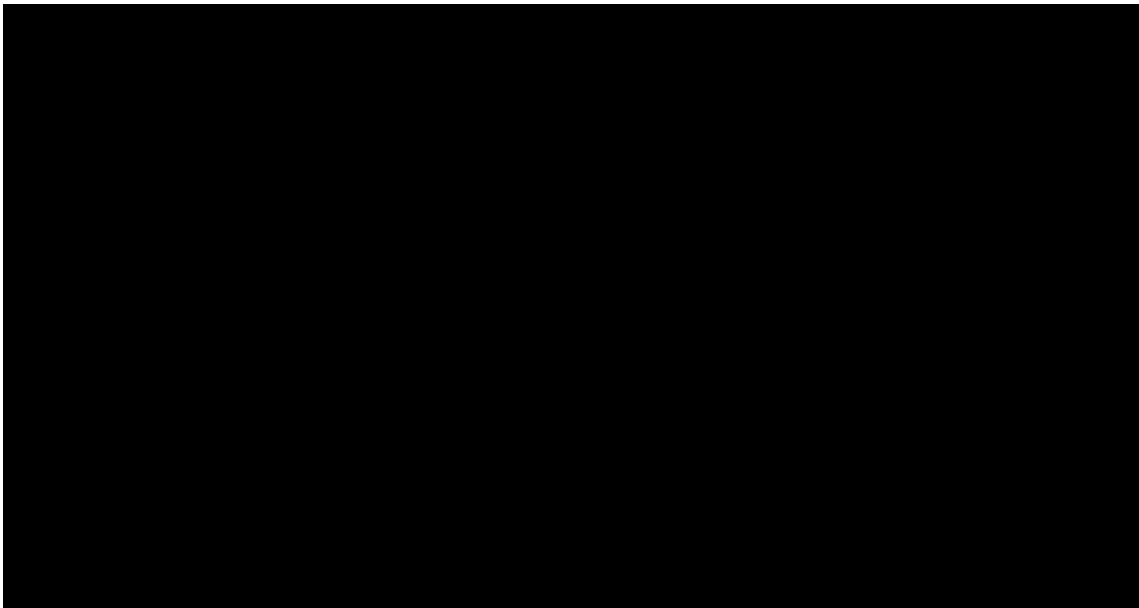


Figura 2.1: Esquema del funcionament del sistema i la interacció amb l'usuari.

En general, el funcionament del sistema sencer es pot descriure amb el flux descrit a la figura 2.1.

En un pas previ de calibració, el sistema obté una imatge del fons (imatge sense cap contacte), que anomenarem de referència. A més, durant aquest pas, se li especifiquen al sistema quines són les mides reals de la pantalla (alçada i amplada) i una sèrie de paràmetres necessaris per distingir quan s'ha fet contacte de quan no.

Durant l'estat de repòs (quan l'usuari no interactua amb el dispositiu), la càmera va recollint imatges i les passa al mòdul de processament d'imatge. Aquest compara la imatge obtinguda amb la de referència. En veure que són iguals, el flux acaba aquí.

Per altra banda, quan l'usuari veu l'escena en la pantalla i decideix interactuar-hi, la interacció comença. Per fer-ho, l'usuari toca la pantalla. A nivell de maquinari, la càmera enregistra la imatge amb el contacte i la passa al mòdul de processament d'imatge. Aquest, novament, compara la imatge que se li passa amb la de referència i detecta que hi ha diferències. Utilitzant els paràmetres obtinguts durant la configuració, determina si les diferències s'han de considerar contacte o no (p. ex: si l'usuari passa la mà per damunt de la pantalla sense tocar-la). En cas afirmatiu, calcularà on s'ha produït contacte i enviarà aquesta informació al seguidor de taques. Aquest determinarà si el contacte rebut és nou o es correspon a un d'anterior que s'ha mogut. En ambdós casos, enviarà la informació a l'aplicació perquè aquesta generi la sortida com a resposta al contacte de l'usuari. L'aplicació actualitzarà l'escena d'acord amb la informació rebuda (nombre de dits i posició d'aquests) i la passarà al dispositiu de sortida (ja sigui una pantalla, un projector o qualsevol altre). En aquest moment, l'usuari pot seguir interactuant amb el sistema, fent que es repeteixi la seqüència anterior, o no seguir, deixant-lo en estat de repòs.

El rendiment dels dispositius multitàctils basats en càmera depèn tant del programari com del maquinari. Quan l'usuari toca el dispositiu, espera que aquest respongui immediatament (o en un interval de temps imperceptible). La velocitat de resposta (responsiveness) del dispositiu ve donada pel temps que necessita per processar l'entrada de l'usuari i per mostrar la resposta per pantalla. Utilitzar una càmera capaç de prendre 30 imatges per segons permet que la interacció sigui fluida. Però al mateix temps, fa necessari un sistema que pugui processar les imatges i generar la sortida a aquesta freqüència. Una combinació dels algorismes implementats en el programari i un maquinari potent minimitzen la latència i augmenten la velocitat de resposta del dispositiu.

2.1 Processament de la imatge

El processament d'imatges és el conjunt de tècniques que s'utilitzen per extreure característiques d'interès de les imatges, com els contorns i les regions, per corregir distorsions geomètriques i per a l'eliminació de soroll d'adquisició.

Aquest és, doncs, el primer pas que cal fer en el procés de detecció del contacte dels dits a la taula. Cada implementació de seguidor de taques per a taules multitàctils

utilitza el seu propi esquema de filtres, però en general, la imatge és processada pels següents mòduls:

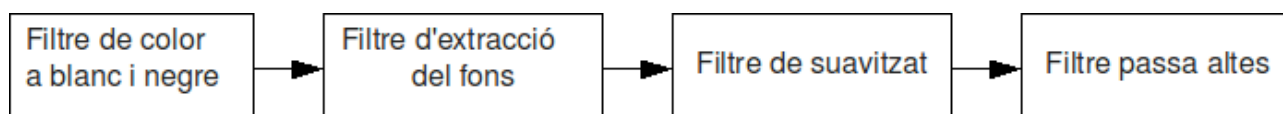


Figura 2.2: Esquema dels mòduls de processament de la imatge

En ser un conjunt d'operacions que s'han de fer per cada imatge captada per la càmera, és necessari que siguin ràpides. Per això s'utilitzen imatges en blanc i negre i es basen en operacions senzilles per a un ordinador.

Així doncs, la cadena de filtres per la que passa cada imatge està formada per:

- Filtre de color a blanc i negre,
- Filtre d'extracció de fons,
- Filtre de suavitzat,
- Filtres de passa altes.

A continuació és descriu cadascun d'aquests mòduls.

2.1.1 Filtre de color a blanc i negre

Aquest filtre converteix la imatge que arriba des de la càmera en color a escala de grisos. Encara que visualment les imatges que arriben des de la càmera semblen en blanc i negre, a nivell intern, s'utilitzen 3 canals per a la seva representació. Aquest filtre converteix aquesta imatge de 3 canals de colors (ja siguin RGB, YUV, o qualsevol altre) a 1 un sol canal, dividint per 3 la quantitat d'informació. D'aquesta manera, la resta de filtres treballaran amb menys informació i seran més ràpids.

2.1.2 Filtre d'extracció del fons

Aquest filtre resta el valor de cada píxel de la imatge del fons (o de referència) al de la imatge obtinguda per la càmera, tal i com es pot veure a la figura 2.3. D'aquesta manera, a la imatge resultant només queden les diferències entre elles (en principi, els dits de l'usuari).

Encara que hi ha mètodes d'extracció dinàmica del fons, els següents filtres fan que l'operació de subtracció de la imatge de referència sigui suficient per obtenir bons resultats.

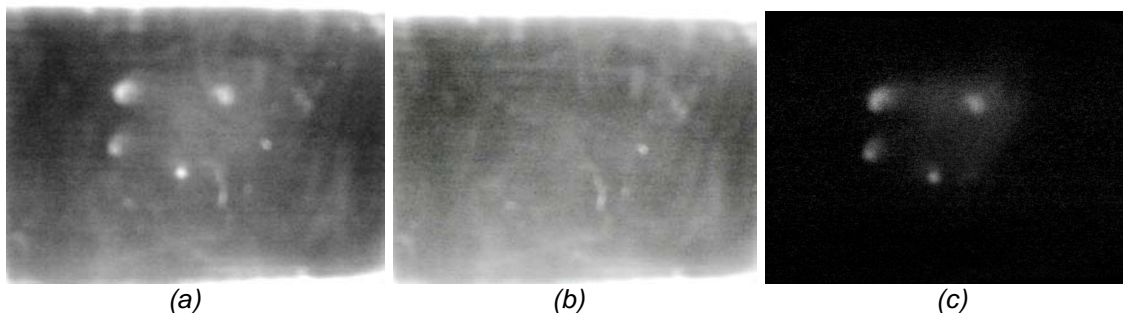


Figura 2.3: a) Imatge que arriba des del filtre de blanc i negre. b) Imatge de referència (o background). c) Imatge resultant del filtre d'extracció del fons.

2.1.3 Filtre de suavitzat

La idea darrere l'operació de suavitzat, o desenfocament, és que el valor dels píxels d'una imatge real presenten diferències petites en variacions petites al llarg de l'espai i, per tant, hi ha una relació entre els píxels veïns. Per contra, el soroll aleatori pot presentar grans diferències entre píxels adjacents i, per tant, no està relacionat amb l'espai. Així, aquest filtre intenta eliminar aquest soroll mantenint la variació original del valor dels píxels (figura 2.4).

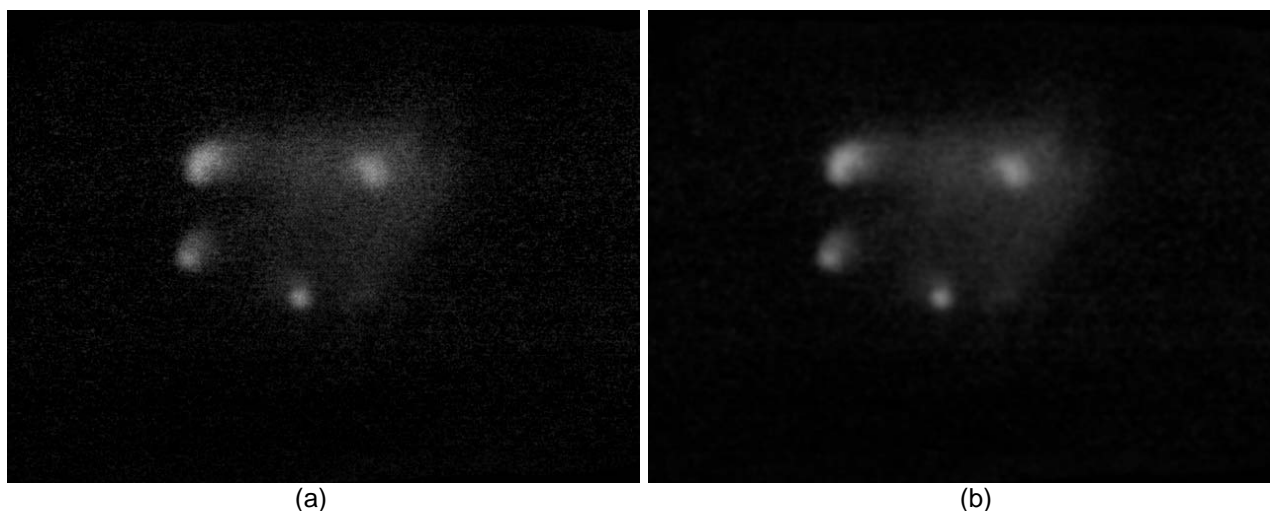


Figura 2.4: Imatge resultant d'aplicar el filtre de suavitzat.

2.1.4 Filtre passa altes

Aquest filtre elimina les parts menys lluminoses que un cert llindar, de manera que a la imatge només queden les taques amb lluminositat més alta que aquest. Aquest procés consisteix a posar negre en els píxels que tenen un valor menor al llindar i posar blanc en els que tenen un valor superior (fig. 2.5). És per això que s'anomena filtre passa altes: només queden aquells píxels amb valor major que el llindar.

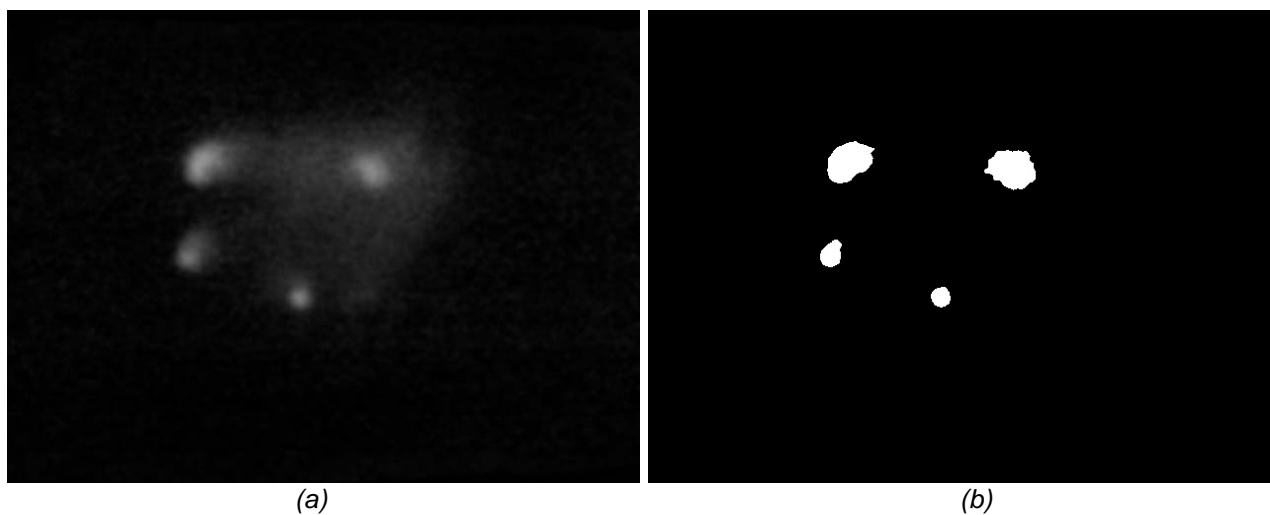


Figura 2.5: Resultat d'aplicar el filtre passa altes.

Per altra banda, aquest filtre permet distingir quan l'usuari fa contacte i quan no.

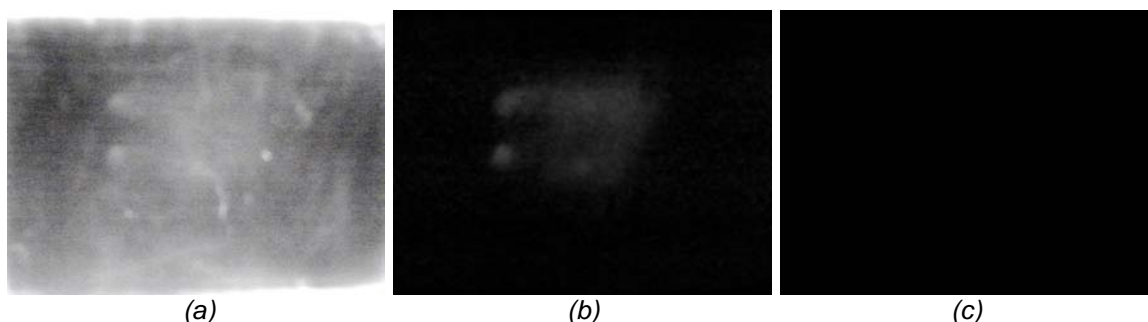


Figura 2.6: a) Exemple d'imatge en la que l'usuari no fa contacte. b) Resultat d'aplicar el filtre d'extracció del fons i de suavitzat a la imatge de la figura (a). c) Imatge resultat d'aplicar el filtre passa altes a la imatge de la figura (b).

A la figura 2.6c) es pot veure com, encara que la imatge resultat de l'extracció del fons i de suavitzat (fig. 2.6b) conté taques, el filtre passa altes les elimina perquè no són prou lluminoses.

2.2 Extracció i seguiment de les taques

El seguidor de taques (de l'anglès *blob tracker*) és el mòdul que s'encarrega de detectar-les (o extreure-les) i seguir-les:

- **L'extracció** consisteix a saber on s'ha tocat la pantalla a partir de les imatges de la càmera.
- El **seguiment** consisteix a relacionar les taques de la imatge d'un instant donat amb les de la imatge de l'instant anterior.

2.2.1 Extracció de la taca

Després de ser tractada pel mòdul de processament d'imatges, la imatge resultant és utilitzada per extreure'n les taques. Per fer-ho, el sistema ha de trobar les coordenades i, depenent del mètode utilitzat, la forma d'aquestes.

Així doncs, un cop es té la imatge de fons negre amb les taques blanques, cal agrupar els píxels blancs segons la taca a la que pertanyen, com es veu a la figura 2.7.

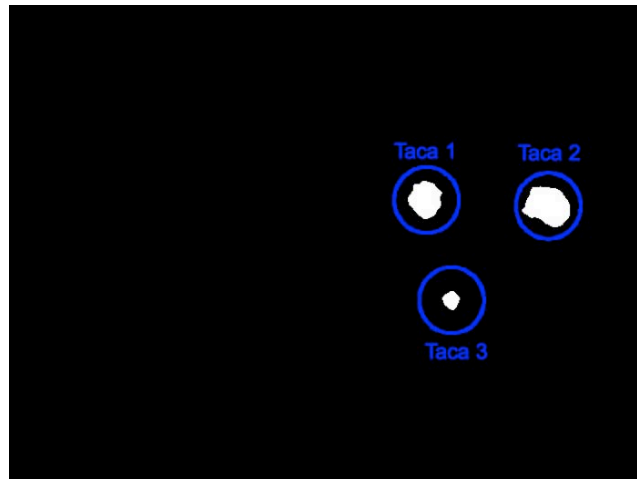


Figura 2.7: Taques marcades corresponents a 3 dits.

Així, es guarda una llista amb les taques trobades i les seves propietats. Aquestes inclouen la posició (normalment del centre), l'orientació i la mida. Aquesta llista és passada al seguidor, que ja només treballarà amb ella i no amb les imatges.

2.2.2 Seguiment dels taques

El seguiment de les taques consisteix a relacionar les taques trobades a la imatge de l'instant t amb les trobades a la de l'instant $t-1$ (fig. 2.8). El cas més senzill és quan hi ha el mateix nombre de taques a les dues imatges. El procés es complica si, a més, poden aparèixer o desaparèixer taques (degut a que l'usuari utilitza més o menys dits per interactuar).

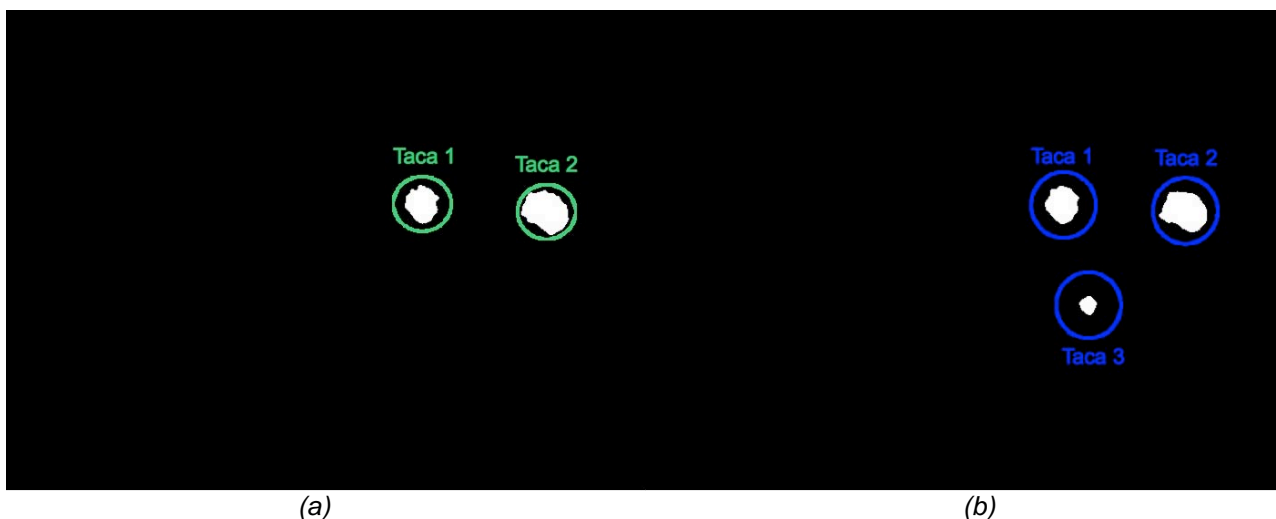


Figura 2.8: Dues imatges consecutives. A la segona, l'usuari utilitza un dit més que a la primera.

2.3 Transmissió de les dades

Un cop el sistema de reconeixement i seguiment té la llista de blobs els envia l'aplicació amb la que està interactuant l'usuari. Tal com s'ha dit anteriorment, l'aplicació amb la que l'usuari interacciona no conté els mòduls explicats en aquest capítol. Aquest enviament es fa utilitzant la xarxa i, per tant, l'aplicació de l'usuari es pot estar executant en la mateixa màquina o en una altra (de manera que un ordinador recull les taques i un altre genera la sortida que veu l'usuari).

Per transmetre les dades referents a les taques, s'utilitza el protocol TUIO [KBB05]. Aquest és un protocol simple però versàtil dissenyat per complir els requeriments de les interfícies d'usuari tangibles. A més, aquest protocol està implementat utilitzant l'OSC (*OpenSound Control*) [WFM03], i per tant el format dels missatges segueix la mateixa sintaxi. Per altra banda, el protocol TUIO només utilitza 3 tipus de missatge. Així els missatges i el seu format són:

```
/tuio/[profileName] set sessionID [parameterList]
/tuio/[profileName] alive [list of active sessionIDs]
/tuio/[profileName] fseq
```

on:

- *profileName* és el nom del perfil (en el nostre cas 2Dobj per a objectes, 2Dcur per a cursors i dits).
- *sessionID*: identifica el frame o estat.
- *parameterList* és la llista de paràmetres referents a l'objecte o cursor. En el nostre cas, aquests són:

- *i*: identificador del dit,
- *x, y*: posició,
- *m*: acceleració del moviment,
- *r*: acceleració de la rotació,
- *list of active sessionIDs* és la llista amb tots els frames que estan actius.

El primer missatge (*set*) indica al client que el cursor *i* s'ha mogut. Així el sistema va enviant aquest missatge periòdicament amb la informació d'aquells cursors que s'han mogut durant el *frame sessionID*, per tal de que el client només es preocupi d'aquests.

El segon missatge (*alive*) envia la llista de totes les sessions que estan actives. Això permet al client saber quins punters encara s'han de considerar com a vius (o en contacte amb la taula) i quins no. Aquest sistema permet estalviar-se un missatge per saber de la creació i de l'eliminació dels cursors: aquesta tasca es deixa al client.

L'últim missatge (*fseq*) s'utilitza com a sistema de sincronització. Quan el client el rep, entén que el sistema de seguiment, a partir d'aquell moment, li envia informació sobre el nou estat dels dits *i*, per tant, cal actualitzar l'escena.

Un exemple del que s'envia quan l'usuari posa dos dit és:

```
/tuio/2Dcur 0 0 0,00208333 0,0015625 0 0 pel punt p1
/tuio/2Dcur 0 1 0,00833333 0,00625 0 0 pel punt p2
```

Per tant, l'aplicació que genera l'escena, i amb la que l'usuari interactua, ha de ser capaç de rebre aquest informació i processar-la. Un exemple de tecnologia que no pot utilitzar directament aquest protocol és *Flash* d'*Adobe*. En aquest cas cal col·locar una aplicació al mig de la transmissió per convertir els paquets OSC sobre UDP en XML sobre TCP (com per exemple FLOSC [Chu]), o utilitzar un *tracker* que envii les dades directament en el segon format.

Tot això implica que l'aplicació amb la que l'usuari interactua ha de ser capaç de rebre aquestes dades, és a dir:

- Si la càmera que detecta els contacte dels dits no està connectada a l'ordinador en que s'executa l'aplicació, aquest ha d'estar connectat a la xarxa.
- L'aplicació haurà de poder crear un fil (*thread*) per tal de rebre les dades i processar-les, independentment del codi de l'aplicació mateixa.

- L'aplicació haurà de tenir el codi necessari per poder interpretar les dades TUIO que li arribin i convertir-les en events.

3 Construcció i implementació

Tal com s'ha comentat a la introducció, per dur a terme aquest projecte s'utilitzarà:

- La idea d'utilitzar el fenomen òptic de la **reflexió total interna frustrada** de J. Han [Han05] per construir la taula (veure capítol 3.1.2).
- El *tracker* **Comunity Core Vision** [CCV] de NuiGroup pel seguiment dels dits (veure capítol 2.1 i 2.2),
- Les llibreries de codi obert del **reactTable** [Ret14] per a la implementació de la recepció de les dades dels dits (veure capítol 2.3).

En aquest capítol es presenta, per una banda, la construcció de la taula multitàctil, amb algunes consideracions i necessitats; i per l'altra, la implementació d'una aplicació d'exemple.

3.1 Construcció d'una taula multitàctil basada en visió per computador

En aquesta secció s'exposa:

- Les consideracions generals a l'hora de dissenyar una taula multitàctil basada en visió per computador,
- Les diferents tècniques que es poden utilitzar per detectar el contacte dels dits amb la superfície utilitzant una càmera,
- La construcció de la taula pròpiament dita.

Així doncs, una taula multitàctil consisteix en una superfície amb la que l'usuari interactua tocant certes zones per donar instruccions i obtenint-hi la resposta, en forma d'imatge. És a dir, la taula serveix tant com a dispositiu d'entrada com de sortida.

Per tal de construir-la, doncs, cal tenir en compte que l'estructura ha de contenir una càmera (per detectar el contacte dels dits), un projector (per mostrar la sortida del programa) i un ordinador (que executi l'aplicació amb la que l'usuari interactua i el *tracker*). A més, cal que hi hagi llum.

És en aquest punt que cal tenir en compte que la sortida del programa utilitza llum (pantalles i projectors) i que la taula i l'usuari estaran en un entorn amb llum ambiental. En utilitzar la mateixa superfície per mostrar la sortida i per captar/sentir el contacte dels dits amb la taula, la càmera també captarà/veurà aquesta llum. Això pot portar/produir problemes a l'hora de distingir el contacte dels dits. Per això, cal "aïllar" la imatge de la

sortida de la imatge de la superfície i els dits en contacte. Per tal de fer-ho, s'utilitza llum infraroja per a la il·luminació i detecció dels dits i llum visible per mostrar la sortida del programa.

3.1.1 Consideracions de disseny

Tal com hem vist, doncs, els elements bàsics que necessitem són (fig. 3.1):

- a. Un projector
- b. Llum
- c. Una càmera
- d. Una superfície plana



Figura 3.1: Esquema conceptual d'una taula multitàctil.

a. Projector

El projector servirà per mostrar l'escena o la sortida del programa a l'usuari. Per això, haurà d'estar col·locat de manera que pugui projectar la imatge en la superfície sense excessives deformacions (encara que els projectors actuals permeten corregir certes deformacions causades pel mal alineament projector-superfície). El millor lloc serà sota de la superfície, ja que si es col·loqués al damunt, les mans i parts del cos podrien fer ombra, privant a l'usuari veure part de la imatge.

Aquí cal tenir en compte que els projectors tenen un cert camp de projecció, depenent de l'angle d'obertura. Així, si es vol obtenir una imatge gran, caldrà situar la superfície de projecció a una distància considerable, que pot arribar a ser d'un metre per a una superfície de 64 x 48 cm (veure figura 3.16). Però aquesta queda limitada per l'alçada de l'usuari, no podent ser major de 80 cm en la majoria dels casos. Per tant, moltes vegades s'utilitza un mirall, que permet dividir la distància projector-superfície en

projector-mirall (normalment horitzontal i limitada per l'àrea de la superfície) i mirall-superfície (normalment vertical i limitada per l'alçada de la taula). Per altra banda, cal tenir en compte que la utilització d'un mirall pot portar problemes d'enfoc, ja que si aquest no està paral·lel a la superfície de projecció, els plans d'enfocament no són els mateixos.

En algunes configuracions s'utilitzen LCD's en comptes del projector. D'aquesta manera, es col·loca l'LCD enganxat a la part inferior de la superfície de contacte i s'utilitza un llum per il·luminar-lo des de sota. Això permet evitar el problema plantejat per la distància projector-superfície, però obliga a tenir l'electrònica de control de l'LCD dins de l'estructura i fa augmentar el preu i la complexitat de construcció.

b. Llum

La funció de la llum, en aquest cas, és permetre al sistema distingir quan l'usuari fa contacte amb la superfície i quan no, a partir de les imatges que recull la càmera. Actualment hi ha dues maneres de distingir aquest contacte: perquè el dit fa ombra (figura 2.2b) o perquè la punta d'aquest queda il·luminada (figura 3.2a).

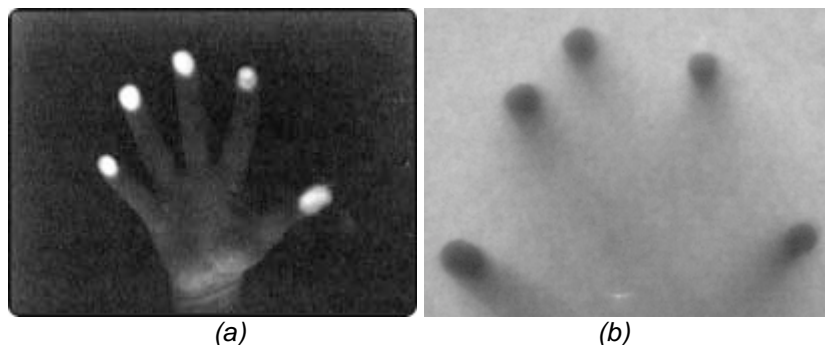


Figura 3.2: a) puntes dels dits il·luminades. b) ombra de les puntes dels dits.

Per tal de que la llum del projector no interfereixi amb la il·luminació dels dits, s'utilitzen dos espectres diferents de llum: llum visible per presentar la imatge a l'usuari i llum infraroja detectar el contacte dels dits. A l'apartat 3.1.2 es descriuen els diferents mètodes d'il·luminació.

c. Càmera

La missió de la càmera, en aquest cas, és captar els dits de l'usuari, o usuaris, en tota la superfície de la taula. A partir de les imatges que aquesta recull, el sistema ha de ser capaç d'extreure'n la posició dels dits. Així doncs, la càmera és col·locarà sota de la superfície de contacte, ja que és el lloc on no hi haurà perill de que els dits es tapin entre ells (com passaria si es col·loqués en un lateral) o que els tapi el palmell de mà (com passaria si es col·loqués al damunt).

Pel que fa a la càmera en si, ens interessarà que tingui una sèrie de característiques concretes per poder obtenir un bon resultat en la extracció dels punts del contacte:

- **Angle de visió:** Depenent d'aquest caldrà ajustar la distància entre la càmera i la superfície de contacte. Encara que és molt important no escurçar-la massa (perquè la càmera pugui veure tota la superfície), no és tant important fer-la més gran del compte, ja que el sistema pot extreure l'àrea d'interès de la imatge. Per contra, com més gran és la distància, menys precisió tindrem. Per això, cal que la càmera tingui una bona resolució. D'aquesta manera, el sensor d'aquesta serà capaç de distingir petits moviments dels dits, encara que estigui a una distància gran de la superfície.
- **Sensibilitat a l'espectre IR:** Tal i com s'ha dit en l'apartat referit a la llum, la detecció del contacte de la punta del dit utilitza llum infraroja. La majoria dels sensors que porten les càmeres digitals (del mòbil, webcams, etc) són capaços de captar els IR, però algunes vegades porten un filtre d'aquest espectre dins de l'òptica, de manera que a la imatge resultant no hi apareixen. Per tant, per poder construir la taula, caldrà utilitzar una càmera d'IR (de vigilància, p. ex) o extreure el filtre d'infrarojos de l'òptica. Al mateix temps, caldrà que la càmera no capti la llum normal, de manera que es pugui separar l'espectre IR per a la detecció i l'espectre normal per a la visualització.

d. Superfície

Donat que el millor lloc per col·locar la càmera i el projector és sota de la superfície, aquesta haurà de ser transparent. Així, el material més utilitzat és el metacrilat. Normalment s'utilitza amb un gruix de 0,5 a 1 cm, depenent del sistema d'il·luminació (veure apartat 3.1.2). Aquest gruix permet a l'usuari fer força pressió sense perill de que es trenqui.

Al mateix temps, però, cal que l'usuari pugui veure la imatge projectada. És per això, que es col·loca un material semitransparent a sota, per tal de no obstruir la llum provinent dels dits. Depenent de la mida de la superfície, es pot utilitzar paper de ceba o plàstic.

La utilització d'aquest material en comptes de vidre es deu a que el vidre pesa més i és menys transparent.

3.1.2 - Mètodes d'il·luminació per a taules multitàctils basades en visió per computador

La comunitat de NUIGroup ha anat recollint els diferents mètodes per detectar el contacte dels dits amb la superfície de la taula. En aquest apartat es presenten aquests mètodes d'il·luminació. L'elecció de cada un d'ells depèn del cost i la complexitat que es pugui assumir en la construcció de la taula i l'entorn en que aquesta es trobi un cop muntada (degut, sobretot, a la quantitat i tipus d'il·luminació d'aquest entorn).

Així, els mètodes exposats són:

- a. Reflexió Total Interna Frustrada (RTIF)
- b. Il·luminació Difusa (ID)
- c. Pla de Llum Làser (PLL)
- d. Il·luminació de Superfície Difusa (ISD)

A continuació es descriuen aquests mètodes i s'exposen alguns dels principals avantatges i inconvenients que tenen.

a. Reflexió Total Interna Frustrada

Aquest mètode aprofita l'efecte de reflexió total interna (RIT o en anglès *Total Internal Reflection*, *TIR*).

La reflexió total interna és un fenomen òptic que té lloc quan un raig de llum incideix en la frontera d'un mitjà en un angle major que l'angle crític respecte la normal a la superfície. Si l'índex de refracció és menor en l'altre costat de la frontera, la llum no pot passar i es reflecteix. L'angle crític és l'angle d'incidència per sobre del qual la reflexió total es produeix [Han05] (fig. 3.3).

Així, el que es fa en aquest tipus de taules és il·luminar la superfície (normalment de metacrilat) amb llum infraroja des dels laterals d'aquesta. Això fa que el metacrilat quedi "inundat" de llum. Tal com es veu a la figura anterior, la llum queda atrapada i, per tant, la càmera no és capaç de veure-la.

Però en el moment en què l'usuari fa contacte amb el dit, les condicions canvien, i es frustra l'efecte RIT, la qual cosa dóna lloc a l'efecte de reflexió total interna frustrada (RTIF o en anglès *Frustrated Total Internal Reflection*, *FTIR*). En aquest cas, la llum, en comptes de tornar a reflexar-se cap a dins del metacrilat de la pantalla, rebota en el dit, sortint del metacrilat i incidint en el sensor de la càmera (fig. 3.4). D'aquesta manera, és possible distingir quan l'usuari fa contacte i quan no.



Figura 3.3: Efecte RIT en polimetacrilat de metil.

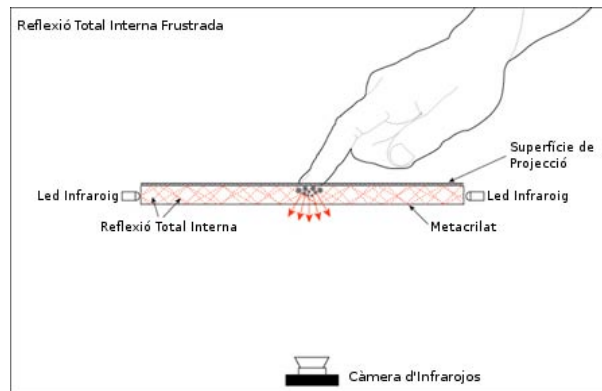


Figura 3.4: Esquema de reflexió total interna frustrada

b. Il·luminació difusa

La il·luminació difusa es presenta en dues configuracions:

1. Il·luminació difusa posterior:

En aquesta configuració, es col·loquen els emissors de llum infraroja a sota de la superfície de contacte, tal i com es veu a la figura 3.5. A més, es col·loca un material difusor en la part superior o inferior de la superfície de manera que, quan un objecte la toca, reflecteix més llum que el difusor i aquesta llum extra és detectada per la càmera. Depenent del difusor, aquest mètode també pot detectar quan es passa la mà per sobre sense fer contacte i permet distingir objectes col·locats en la superfície.

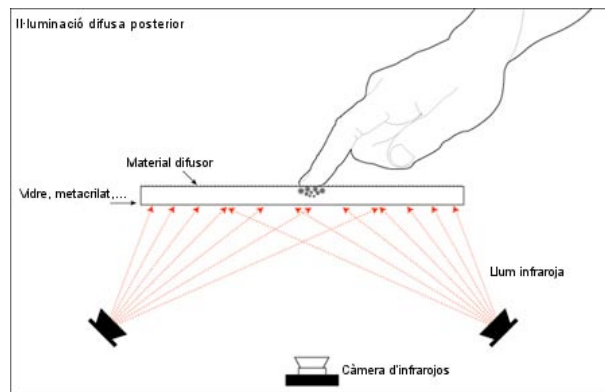


Figura 3.5: Esquema d'il·luminació difusa posterior

2. Il·luminació difusa frontal:

En aquest cas, la llum infraroja arriba per damunt de la superfície de contacte. Aquesta llum pot ser la llum ambient que envolta la taula o pot sortir d'un focus utilitzat per a tal efecte. Com en el cas anterior, s'utilitza un material difusor col·locat a la part superior o inferior de la superfície, però quan un objecte toca la superfície, en comptes de quedar il·luminat, projecta una ombra que és captada per la càmera.

c. Pla de llum làser

Com el seu nom indica, aquest mètode utilitza làsers com a font de llum infraroja. Aquests làsers es col·loquen de manera que projecten la llum paral·lela a la superfície. El gruix del feix, normalment, és d'1 mm. Quan l'usuari fa contacte, la punta del dit queda il·luminada i la càmera pot detectar-la, tal i com es mostra a la figura 3.6.

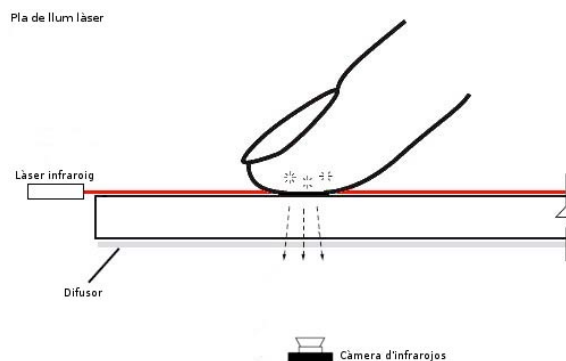


Figura 3.6: Esquema de pla de llum làser

d. Il·luminació de superfície difusa

La ISD (en anglès *Diffused Surface Illumination*, o DSI) utilitza un tipus especial de metacrilat per distribuir uniformement la llum infraroja a través de la superfície. Aquest material conté petites partícules en el seu interior que actuen com a milers de petits miralls. Quan la llum d'infrarojos brilla des de les vores d'aquest material, la llum és

redireccionada i propagada cap a la superfície (fig 3.7). Bàsicament, té una configuració com el *FTIR* però amb el metacrilat especial.

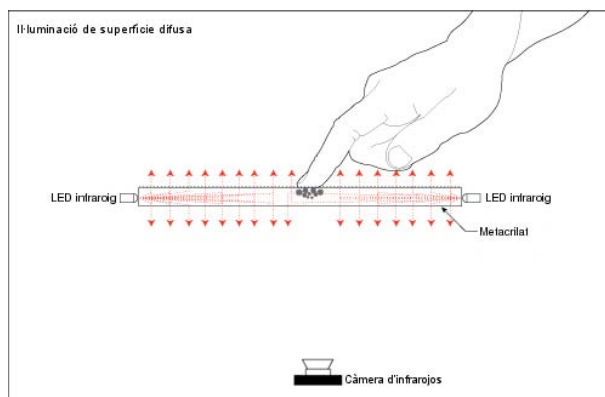


Figura 3.7: Esquema d'il·luminació de superfície difusa

En la taula 2 es presenta una comparativa d'algunes de les característiques, de caire pràctic, més importants de cada mètode.

| | Contrast | Detecta pressió | Reconeix objecte o fiduciaris | Complexitat de construcció | Cost |
|--------------|----------|-----------------|-------------------------------|----------------------------|------|
| RITF | Alt | Sí | No | Alta | Alt |
| ID posterior | Baix | No | Sí | Baixa | Mig |
| ID frontal | Baix | No | No | Baixa | Baix |
| PLL | Alt | No | No | Mitja | Mig |
| ISD | Baix | Si | Sí | Alta | Alt |

Taula 2: Comparació dels diferents mètodes d'il·luminació

3.1.3 Construcció d'una taula multitàctil basada en visió per computador

En aquest apartat es presenta el procés de construcció d'una taula multitàctil basada en visió.

S'han construït dues taules: una de prototipat per analitzar les necessitats, amb costos reduïts, i una de definitiva. En aquest apartat es presenten totes dues. El mètode d'il·luminació escollit és el de RITF. La principal raó per utilitzar aquest mètode és l'alt contrast que ofereix, fent que els dits apareguin molt il·luminats. D'aquesta manera, és més fàcil distingir el contacte d'aquests en la superfície. Per altra banda, ofereix una bona independència respecte la llum ambiental, de manera que un cop s'han calibrat els filtres del *tracker*, es pot moure la taula de lloc o es pot canviar la llum del seu entorn sense que això afecti a la correcta detecció del contacte. Finalment, com que no utilitza làsers, no planteja cap mena de perill i per tant es pot utilitzar amb menors sense por de llesions oculars.

a. Taula prototipus

La principal raó per construir aquest prototipus ha estat poder provar el mètode d'il·luminació reflexió interna total frustrada (materials, tècniques i components) sense haver d'invertir molts diners. A més, s'ha fet d'una mida reduïda per poder-la modificar sense que sigui molt problemàtic. Així doncs, per construir el prototipus s'ha utilitzat:

- Metacrilat de 32 cm x 24 cm x 0.6 cm com a superfície de contacte.
- *Leds* de llum IR de 850 nm de longitud d'ona, per il·luminar la superfície.
- Perfils d'alumini en angle de 2 cm x 2 cm per muntar l'estructura que aguanta els leds i el metacrilat.
- Fustes per mantenir l'estructura d'al·lumini.
- Paper de ceba com a material de projecció.
- Càmera Sony PS3Eye modificada sense filtre d'IR i amb negatiu fotogràfic velat.

Al ser la primera taula que es construïa, el metacrilat es va comprar sense polir els laterals. Com que la llum ha d'incidir dins d'aquest, cal que estiguin polits i quedin transparents. Així doncs, es van polir utilitzant un raspall de cotó, amb adaptador per trepant i poliment líquid amb silicones.

Una altra conseqüència de ser la primera taula va ser l'elecció dels leds infrarojos. Inicialment se'n van utilitzar amb una longitud d'ona de 940 nm, però es va comprovar que eren massa poc lluminosos i la càmera no era capaç de distingir el contacte dels dits. Finalment s'han utilitzat *leds* de 850 nm.

Donada la brillantor que presenten els *leds* de 850nm, es va decidir col·locar-ne, només, als dos laterals llargs de la pantalla, a una distància de 1,5 cm entre ells. Així doncs, per il·luminar els dos laterals de 32 cm calen $32 \cdot 2 / 1,5 = 43$ *leds*. Finalment, per arrodonir se n'han col·locat 40.

Utilitzant una font d'alimentació de 9V, els 40 *leds* es poden dividir en 8 línies en paral·lel de 5 *leds* en sèrie cada una. Així cada línia necessita una resistència de 51Ω. Es pot veure un esquema del circuit la figura 3.8.

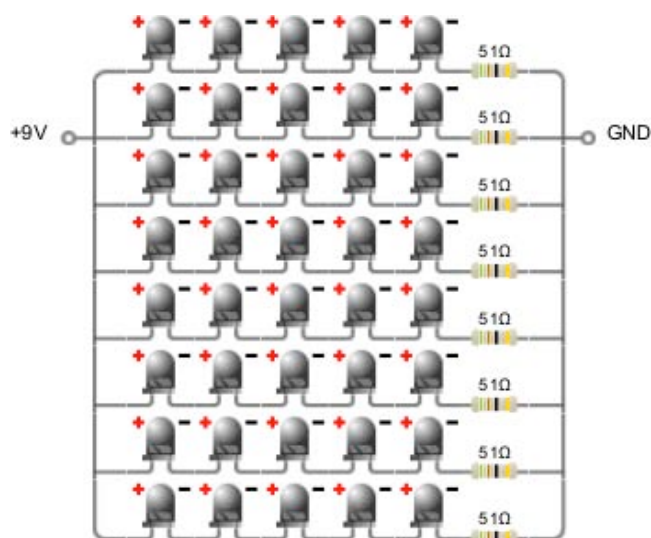


Figura 3.8: Esquema del circuit de leds [LeC]



Figura 3.9: detall d'una de les línies de 5 leds en sèrie.

A més a més, sota el metacrilat, cal col·locar un material semi-transparent, capaç de mostrar la imatge projectada i, al mateix temps, capaç de deixar passar la llum dels dits que fan contacte. Com que és una superfície relativament petita, s'hi va col·locar paper de ceba, obtenint força bons resultats. De cara a la definitiva però, va quedar clar que no es pot utilitzar el mateix material, donada la seva poca rigidesa.

Per construir l'estructura que manté els leds, el metacrilat i el material de projecció en la seva posició, s'han utilitzat perfils d'alumini. Els leds s'han muntat en forats al llarg de l'alumini de manera que quedin fixes perpendicularment al metacrilat, tal i com es pot veure a les figures 3.9 i 3.10. Aquests perfils, al seu temps, s'han muntat sobre un suport de fusta.



Figura 3.10: Estructura per aguantar els leds i el metacrilat.

Per provar que realment funcionés i poder veure l'efecte FTIR es va utilitzar una càmera USB Sony PS3Eye. Aquesta càmera porta de fàbrica un filtre de llum infraroja. Per poder-la utilitzar es va extreure aquest filtre. Després es va utilitzar un negatiu fotogràfic velat com a filtre de llum normal (figura 3.11). D'aquesta manera, la llum ambiental no hauria d'afectar la detecció del contacte. Malauradament, amb aquest mètode no es van obtenir resultats gaire bons (figura 4.5). Això és degut a que el negatiu velat impedeix que passi gran part de llum visible, però no la filtra totalment, fent que la càmera capti llum molt brillant i el sistema no sigui capaç de distingir el contacte dels dits de la llum visible ambiental (com pot ser un focus damunt de la taula).



Figura 3.11: Càmera Sony PS3Eye amb el negatiu.



(a) (b)
Figura 3.12: Prototipus ja acabat. a) Vista frontal. b) Vista posterior

Un cop construït el prototipus es van fer proves per veure si realment funcionava, és a dir, la càmera era capaç de distingir el contacte dels dits.

Definitiva

Un cop construït el prototipus i trobats la majoria de problemes i solucions, s'ha construït una versió definitiva de la taula. En aquest cas ja s'han utilitzat materials de més qualitat i més grans.

Les principals característiques que es desitjaven en el moment de dissenyar-la han estat:

- Que fos utilitzable i funcional.
- Que fos muntable i desmuntable per al seu transport.

Els materials i components que s'han utilitzat per contruir la taula han estat:

- Projector Epson EMP-600 per mostrar la sortida del programa,
- Càmera Sony PS3Eye modificada amb filtre passa-banda de 850 nm per detectar el contacte dels dits,
- Metacrilat de 66 cm x 50 cm x 1 cm com a superfície,
- Cinta de *leds* IR amb longitud d'ona de 850 nm per il·luminar la superfície
- Fustes per muntar l'estructura
- Mirall per poder col·locar el projector en posició horitzontal,
- Plàstic semi-transparent de 0.8 mm com a material de projecció.

Es poden consultar les principals característiques dels components a l'annex A.

La primera part ha estat construir el marc que conté els *leds* i el metacrilat. En aquest cas, s'h utilitzat una cinta de leds prefabricada [EnvL] (figura 3.13). L'ús d'aquesta

cinta ha estat motivat pel cost temporal de soldar tots els leds individuals i de construir l'estructura d'alumini per mantenir-los a lloc.



Figura 3.13: Detall de la cinta de leds.

Així, es va fresar la fusta que serveix de suport de la pantalla de manera que hi encaixessin el metacrilat i els leds. En aquest pas calia tenir en compte que les cantonades no podien ser de 90° ja que la cinta de *leds* es podia malmetre (Figura 3.14). Per altra banda, el metacrilat s'ha tallat amb una mida de 66 cm d'amplada per 50 cm d'alçada. D'aquesta manera, es pot utilitzar un centímetre per cada costat per aguantar-lo i queda una àrea de 64x48 cm per a la visualització, mantenint una relació d'aspecte de 4:3.



Figura 3.14: detall de la cantonada (vista inferior de la taula).

Com a material de projecció, no s'ha pogut utilitzar el paper de ceba, com al prototipus. Això és així perquè aquest material no és gaire rígid i la zona central no queda en contacte amb el metacrilat. Per això s'ha escollit un plàstic semitransparent. A la figura MM es pot veure aquest material abans de ser tallat i lligat amb el metacrilat. A la figura M2 es pot veure el plàstic ja col·locat en la seva posició.

Un cop col·locada la cinta de leds amb el metacrilat i el material de projecció, es van col·locar uns llistons de manera que el metacrilat no caigués, deixant espai per al connector dels leds (figura 3.15). A la figura 3.18 es pot veure la part superior de la taula ja acabada.



Figura 3.15: connector dels leds i marc inferior (fusta més fosca) per aguantar el metacrilat (vista inferior de la taula).

Després es va construir l'estructura que conté el projector, el mirall i la càmera. En aquesta part s'ha hagut de tenir en compte que, per projectar una imatge de 64 d'amplada per 48 cm d'alçada, l'objectiu el projector ha d'estar, aproximadament, a una distància de 92 cm de la superfície sobre la que es projecta.

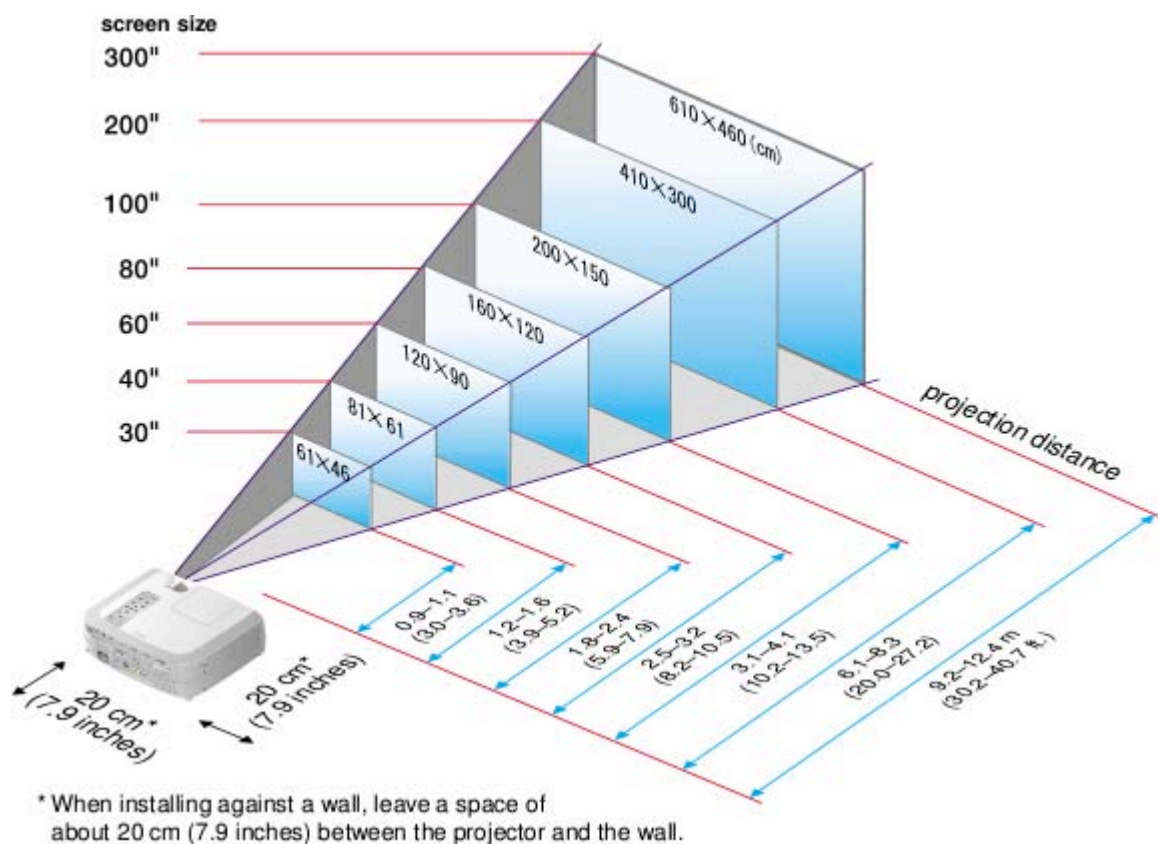


Figura 3.16: Especificacions de distàncies del projector Epson EMP-600.

A més, el projector fa 27 cm entre la part posterior i la davantera. Això implica que si es vol una imatge de 64x48 cm, la superfície de la taula ha d'estar a $92 \text{ cm} + 27 \text{ cm} = 119$

cm de la part posterior del projector, que és una alçada massa gran per a la majoria d'usuaris. Per tal de poder baixar una mica la taula, es va decidir de col·locar un mirall, de manera que el projector estigui en una posició horitzontal.

Durant la col·locació del mirall, cal tenir en compte que, si aquest està en una posició inclinada i la superfície de la taula està en una posició horitzontal, poden aparèixer problemes d'enfocament, donat que els plans no són paral·lels. Així, s'obtenen millors resultats si la superfície de la taula també es col·loca en posició inclinada. Això també permet a l'usuari treballar en una posició més còmode i li facilita l'accés a tota la superfície (Figura 3.20).



Figura 3.17: estructura per a càlcul de distàncies projector-mirall i mirall-superfície, amb el projector en posició horitzontal.

Atesa la dificultat de calcular amb precisió l'angle amb el què havia d'estar col·locat el mirall, assegurar la distància entre aquest i el projector i situar de la càmera per a un correcte enquadrament, s'ha optat per muntar l'estructura amb *Meccano*. Com es pot veure a la figura 3.19b), utilitzant el *Meccano* ha estat possible assegurar una correcta posició dels diferents elements, permetent, al mateix temps, aplicar-hi correccions en cas de ser necessari (moviments a causa del transport o d'altres).

A la figura 3.19a) també es pot veure que la utilització de *Meccano* ha permès fixar la càmera en una posició inclinada a pesar de la seva difícil forma.

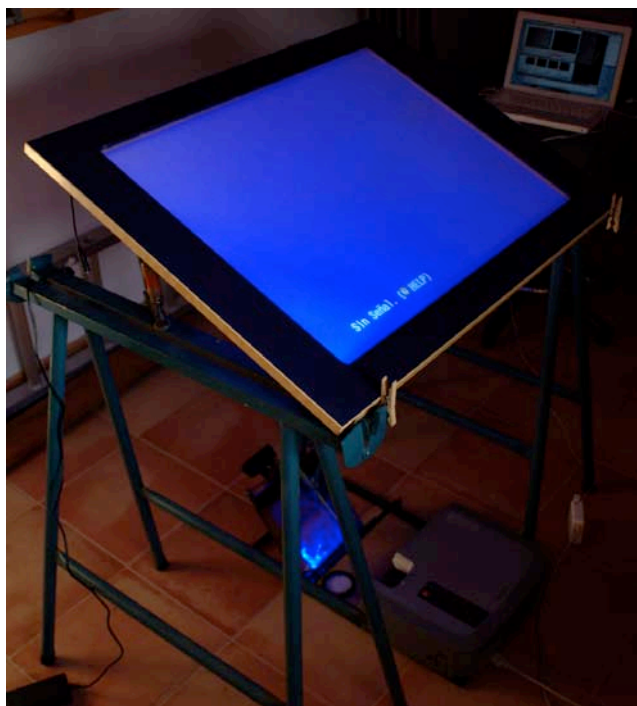


Figura 3.18: Col·locació inclinada de la superfície de la taula.

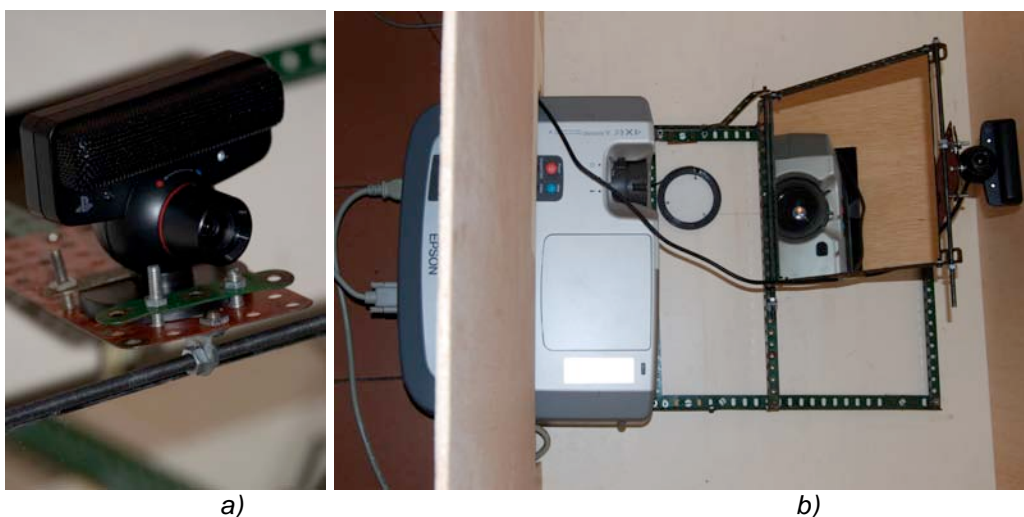


Figura 3.19: a) Detall ancoratge de la càmera. b) Estructura del canó, el mirall i la càmera.

Finalment, s'ha construït l'estructura que aguanta la superfície d'interacció i manté alineades aquesta i la base amb el projector i la càmera. En aquesta estructura s'ha obert una petita porta per tal de poder accedir a l'interior i poder engegar i apagar el projector i ajustar el mirall i la càmera. A més, aquesta porta permet que l'aire calent del projector no s'acumuli i el pugui malmetre.



Figura 3.20: Taula definitiva acabada. a) Vista frontal. b) Vista lateral

3.2 - Implementació

A més de la construcció d'una taula multitàctil, s'ha implementat l'adaptació d'un programa ja existent per tal que permeti una interacció multitàctil. El programa adaptat és l'entorn de pràctiques de l'assignatura Gràfics per Computadors 2 [JGM08]. Aquest entorn ofereix una finestra OpenGL dins d'una de Windows. L'entorn permet visualitzar i modificar objectes 3D utilitzant menus i botons de Windows. L'entorn està programat en Microsoft Visual C++ 2005 i utilitza les llibreries MFC, a més de les OpenGL.

L'entorn, doncs, es pot considerar com un visualitzador d'objectes 3D, permetent:

- Carregar objectes 3D,
- Visualitzar-los des de diferents perspectives i distàncies,
- Transformar-los ja sigui rotant-los, traslladant-los o escalant-los,
- Il·luminar-los segons les diferents il·luminacions OpenGL

L'acció a dur a terme depèn del botó del ratolí que s'estigui utilitzant i del valor de les variables de control. Així, si la variable que controla el *zoom* està activada, quan l'usuari manté pressionat el botó esquerre i mou el ratolí, la resposta del programa serà acostar, o allunyar, l'objecte. Això implica que l'usuari ha de modificar les variables de control (per exemple: a través d'un menú) per accedir a les diferents funcionalitats.

Així, en un principi, l'entorn només permet la interacció utilitzant el ratolí i el teclat. Al ser una aplicació de Windows, és aquest l'encarregat de rebre els events des dels dispositius físics i enviar-los al programa. Un cop rebuts, el programa crida les funcions associades a l'event (per exemple: canviar el color del fons) i modifica les variables de

control (per exemple: punt de vista de la càmera) que defineixen la sortida que veu l'usuari, tal i com es veu a la figura 3.21.

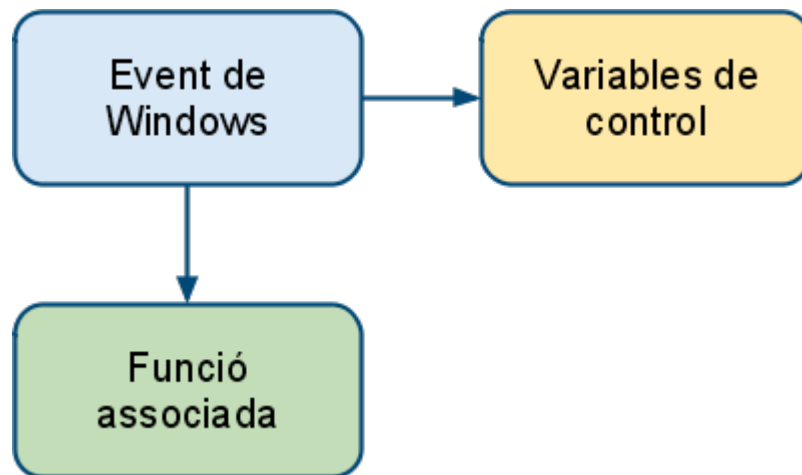


Figura 3.21: Funcionament inicial de l'entorn

Per tal que l'entorn permeti la interacció multitàctil, cal que sigui capaç de rebre els events TUIO. A més, cal modificar les funcions associades, o crear-nes de noves, ja que el tractament d'aquests events no és el mateix que el dels events de ratolí. Tal i com es veu a la figura 3.22, els events TUIO no substitueixen ni modifiquen els de Windows.

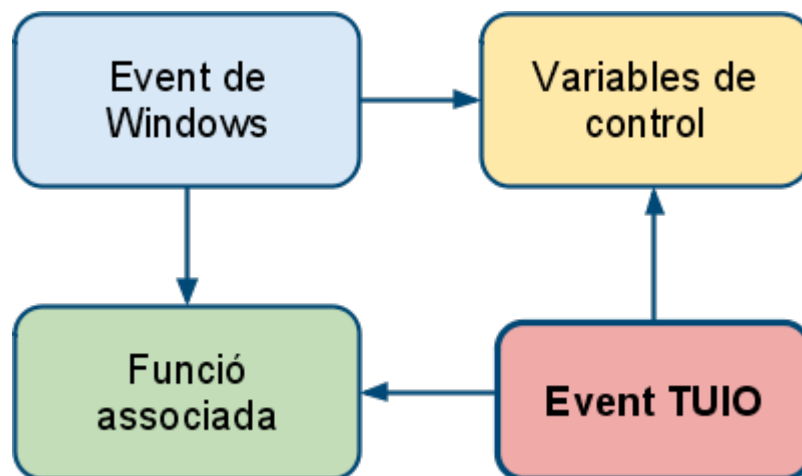


Figura 3.22: Funcionament de l'entorn adaptat

Per tot això, l'adaptació del programa es pot resumir en dues parts:

- a. Recepció de les dades TUIO
- b. Funcions associades i variables de control

a. Recepció de les dades TUIO

Tal i com s'ha explicat en el capítol capítol 2, el *tracker* envia les dades de les posicions dels dits a través de la xarxa utilitzant el protocol TUIO. Per tal de que aquestes dades puguin ser processades pel programa, cal que tingui el codi per rebre-les a traves

de la xarxa i el codi necessari per processar-les. A la figura 3.2.3 es pot veure un diagrama simplificat de les classes utilitzades.

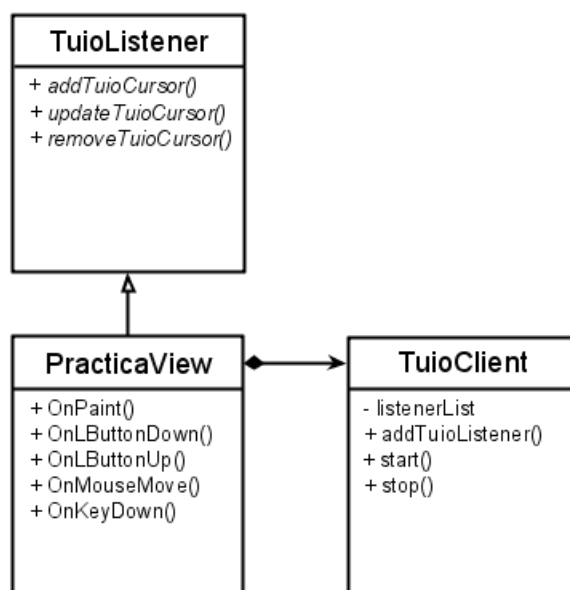


Figura 3.23: Diagrama de classes simplificat per a la recepció i tractament dels events TUIO

El programa original té implementada la classe *PracticaView*, on guarda les variables de control de la finestra OpenGL i on gestiona els events. En la versió modificada, s'han afegit les classes *TuioClient* i *TuioListener*. La primera s'utilitza com a variable membre de *PracticaView*, i és l'encarregada de crear el socket necessari per rebre les dades dels events. La segona, de la qual es deriva *PracticaView*, només declara les funcions abstractes associades als events rebuts. D'aquesta manera, la variable de tipus *TuioClient* pot cridar les funcions de *PracticaView* relacionades amb els events TUIO.

El funcionament d'aquesta part és, doncs:

1. Creació de l'objecte *TuioClient* amb el port que es vol obrir.
2. Afegir l'objecte *PracticaView*, derivat de *TuioListener*, a la llista *listenerList* de l'objecte *TuioClient*.
3. Crida de la funció *start()* de l'objecte *TuioClient* perquè comenci a escoltar i processar missatges TUIO.
4. Per cada missatge que l'objecte *TuioClient* rep a través de la xarxa:
 - L'objecte *TuioClient* mira si el missatge és d'un nou dit. En cas de ser-ho, crida la funció *addTuioCursor()* de tots els *TuioListener's* de la llista *listenerList*. Si no ho és,
 - L'objecte *TuioClient* mira si el missatge és d'un dit ja existent. En cas de ser-ho, crida la funció *updateTuioCursor()* de tots els *TuioListener's* de la llista *listenerList*. Si no ho és,

5. L'objecte *TuioClient* mira si algun dels cursors ha caducat i en cas afirmatiu, crida la funció *removeTuioCursor()* de tots els *TuioListener*'s de la llista *listenerList*.

Els punts 4 i 5 es van repetint fins que es crida la funció *stop()* de l'objecte *TuioClient*.

b. Funcions associades i variables de control

En aquesta segona part, s'ha implementat el codi de les funcions **TuioCursor*, derivades de la classe *TuioListener*.

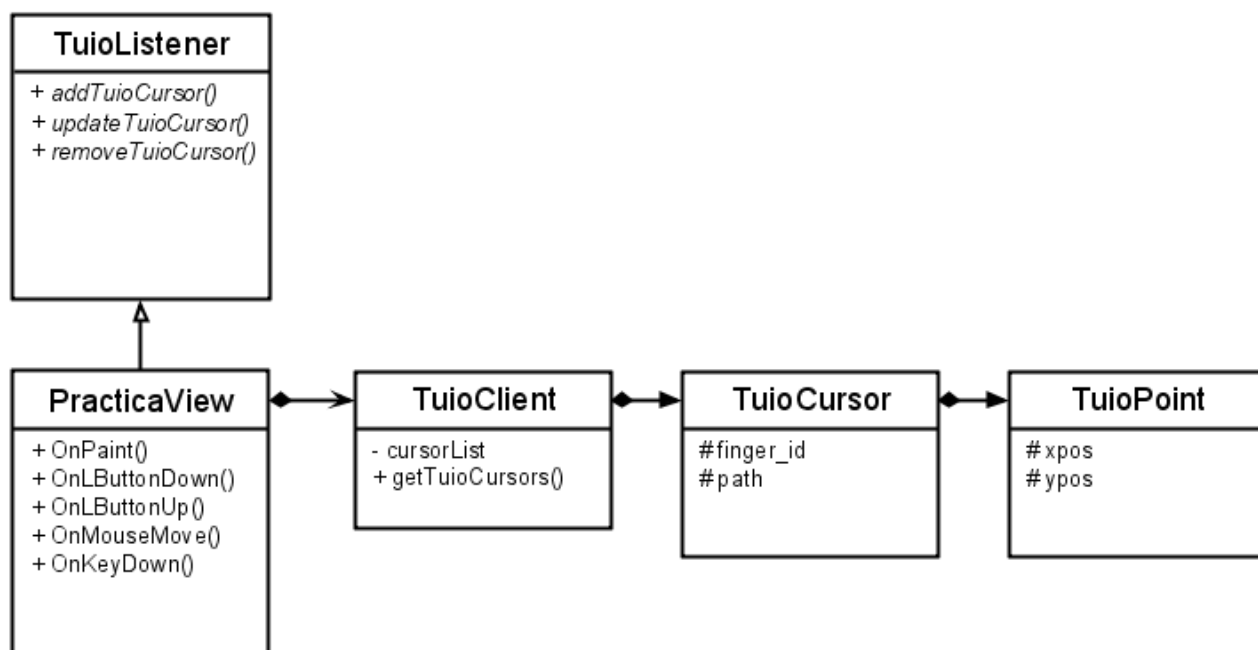


Figura 3.24: Diagrama simplificat de les classes utilitzades per representar els dits i els events generats.

A la figura 3.2.4 es pot veure un el diagrama simplificat que representa un altre punt de vista del codi.

La representació dels dits utilitzada pel programari es fa amb les classe *TuioPoint* i *TuioCursor*. La primera simplement guarda les coordenades x i y del punt. La segona, bàsicament, té un identificador del dit (*finger_id*) i una llista de *TuioPoint*'s pels que ha passat el dit (*path*). Així, es crea una instància de *TuioCursor* quan el dit és detectat per primer cop: se li assigna un identificador únic (ascendent en ordre d'aparició) i s'afegeix el punt d'aparició a la llista *path*. Aquesta instància es va actualitzant (s'afegeixen punts a la llista *path*) mentre es va movent. En el moment en que s'aixeca el dit, es destrueix.

La classe *TuioClient*, en aquesta part, s'utilitza per gestionar els objectes *TuioCursor*. Quan l'objecte *TuioClient* rep informació d'un nou dit, afegeix un nou objecte *TuioCursor* a la llista de cursors. Així, en tot moment es pot saber quants dits hi ha tocant la taula, i la seva informació (posició, camí, etc). A més, com s'ha comentat en l'apartat

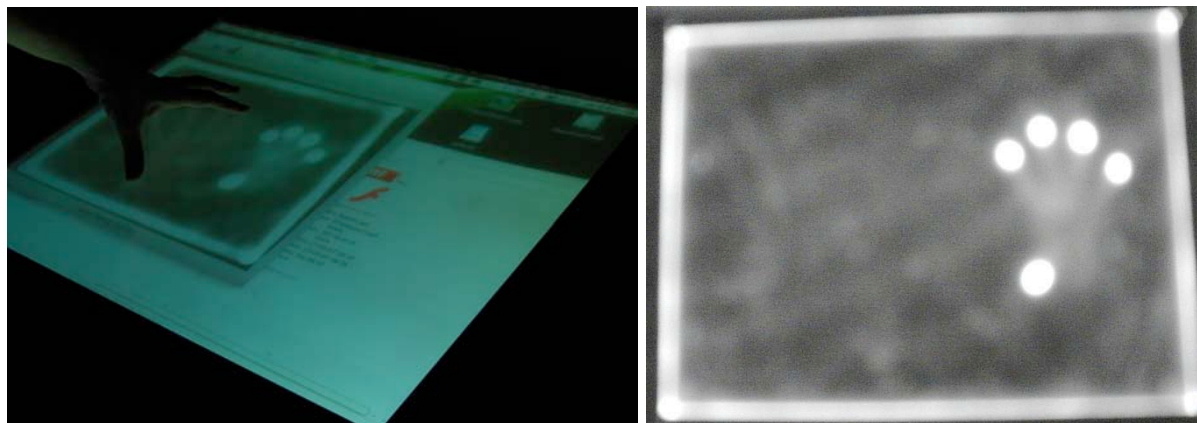
anterior, aquesta classe és l'encarregada de cridar les funcions **TuioCursor* de la classe *TuioListener*.

Finalment s'han implementat les funcions **TuioCursor* de la classe *PracticaView* (derivades de *TuioListener*). Aquestes funcions són les encarregades de modificar les variables de control per tal que la sortida del programa sigui consistent amb les accions de l'usuari. Un exemple d'aquest codi és el que gestiona l'actualització dels dits (*updateTuioCursor*). Quan es crida aquest codi:

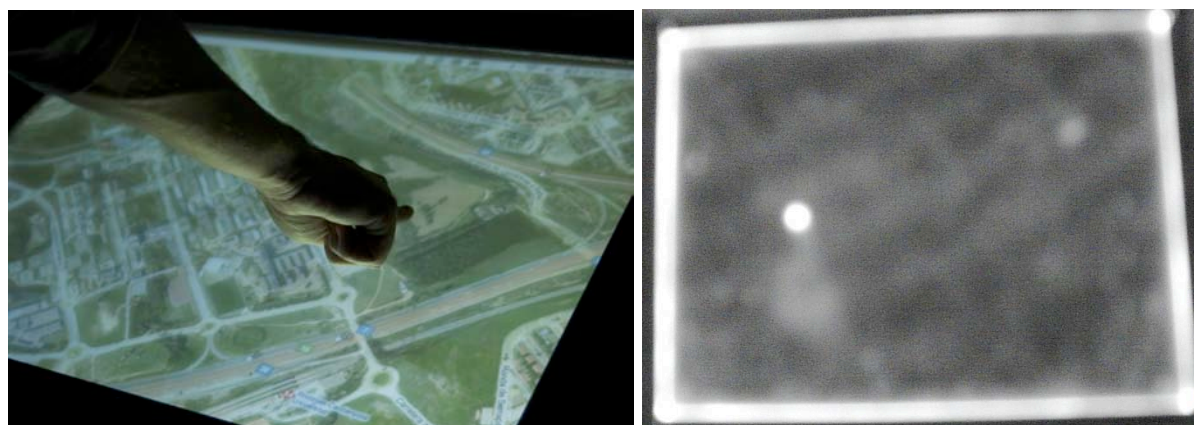
- Mira quants dits hi ha en contacte a partir de la llista de dits *cursorList* de la classe *TuioClient*.
 - Si n'hi ha un, desplaça la càmera (o el punt de vista de l'usuari) segons el moviment del dit (es pot obtenir a partir de la llista *path* del *TuioCursor* que representa el dit).
 - Si n'hi ha dos, allunya o apropa la càmera depenent de si la distància entre els dits augmenta o disminueix.
 - En qualsevol altre cas, no fa res.

4 Resultats

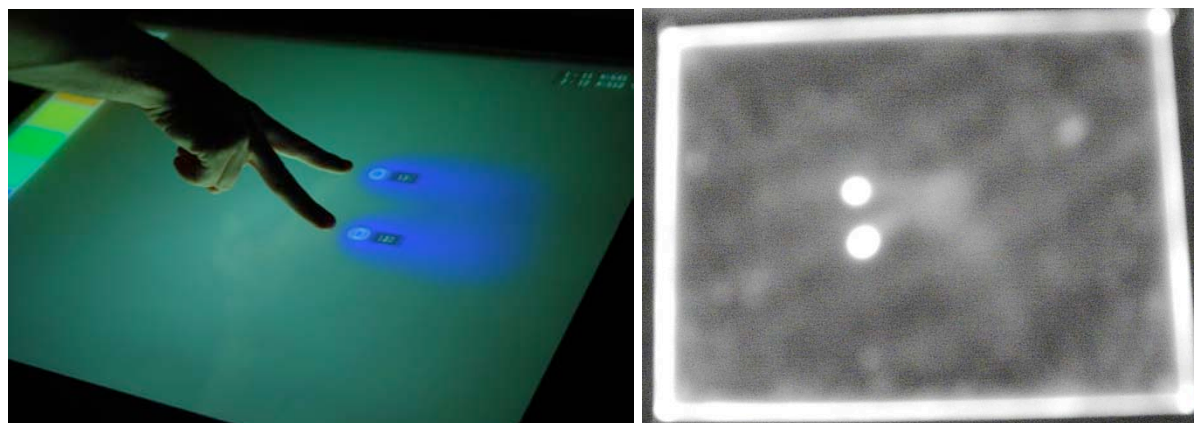
En aquest capítol es alguns els resultats obtinguts. Les primeres 4 figures corresponen a experiments amb usuaris i es presenta el que aquest veu i el que la captura de la càmera. La figura 4.5 mostra la diferència de filtrat entre el filtre passa banda utilitzat en la taula definitiva i el negatiu vel-lat del prototipus (veure capítol 3.1.3).



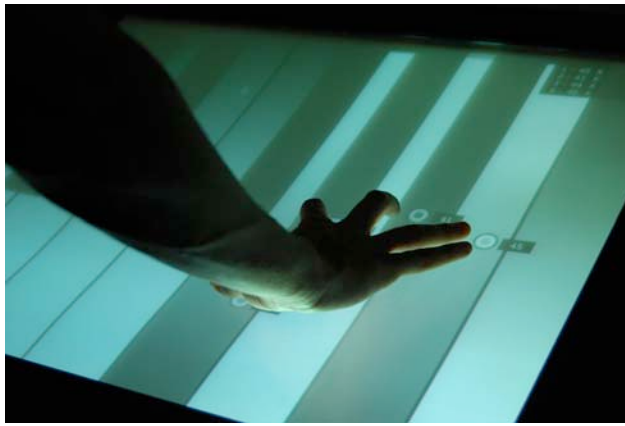
a) b)
Figura 4.1: Col·locació de 5 dits damunt de la pantalla.



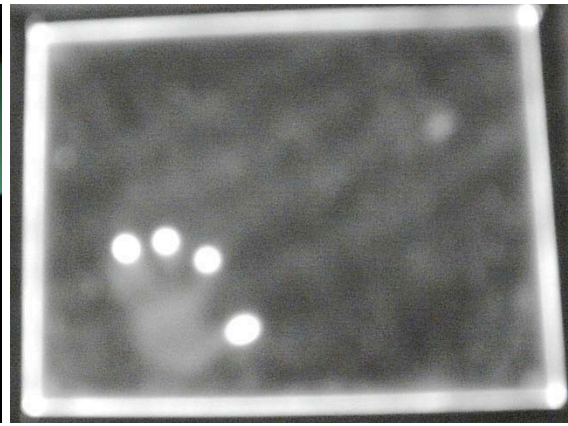
a) b)
Figura 4.2: Utilització d'un dit per navegar per un mapa.



a) b)
Figura 4.3: Utilització de dos dits per pintar.



a)

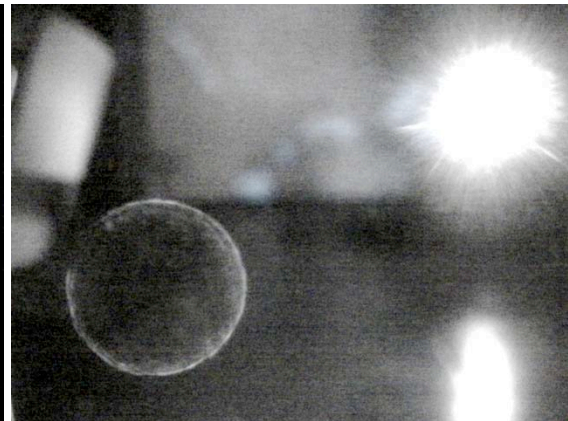


b)

Figura 4.4: Col·locació de 5 dits damunt de la pantalla.



a)



b)

Figura 4.5: Proves del filtre de la càmera amb una llanterna. a) Filtre passa banda de 850 nm. b) Negatiu vel·lat.

5 Conclusions i millores

A continuació s'exposen els principals punts duts a terme en aquest projecte:

- S'ha estudiat el processament d'imatges que permet la detecció del contacte dels dits sobre la taula a partir de les imatges preses per una càmera.
- S'ha estudiat el funcionament de la llibreria del reactTable per a la recepció de les dades TUIO i la generació dels events corresponents.
- S'han estudiat els diferents mètodes d'il·luminació que permeten la detecció del contacte dels dits mitjançant una càmera. D'aquests s'ha escollit el de Reflexió Interna Total Frustrada per a la construcció de la taula multitàctil. Aquest dona independència a la il·luminació ambiental, de manera que un cop calibrat el tracker, la taula seguirà detectant el contacte encara que la taula es canviï de lloc. A més, aquest mètode fa que els dits quedin molt il·luminats, facilitant la detecció del contacte.
- S'ha dissenyat un primer prototipus de taula multitàctil basada en visió per computador. Un cop dissenyat s'ha contruït de manera que només permet la detecció del contacte dels dits i no permet la projecció de la imatge. Amb aquest prototipus s'ha comprovat que s'obtenen millors resultats amb leds IR de 850 nm de longitud d'ona que amb els de 940. Per altra banda, s'ha comprovat que un negatiu vel·lat no filtra prou la llum visible, de manera que no és possible utilitzar-lo.
- S'ha dissenyat una taula multitàctil completa de manera que l'estructura conté el projector i la càmera, a més de la superfície de contacte. Un cop contruïda s'ha comprovat que funciona correctament. La utilització d'un filtre passa banda de 850 nm, instal·lat a la càmera, ha demostrat molt bons resultats. L'ús d'un mirall ha permès fer-la d'una alçada còmode per a la majoria d'usuaris. Per altra banda, s'ha comprovat que aquest mirall (que conté un vidre) provoca la projecció de dues imatges.
- S'ha testejat la taula definitiva demanant a diferents usuaris que la fessin servir per tasques de visualització d'imatges, navegació per un mapa i dibuix amb els dits, entre d'altres. Això ha permès ajustar els filtres del processament d'imatge per a una interacció còmode.

- S'ha implementat l'adaptació d'un programa ja existent [JGM08] perquè permeti la interacció amb més d'un dit simultàniament. Per fer-ho, s'han utilitzat les llibreries del `reactTable` per a la recepció de les dades TUIO. Així, utilitzant un sol dit es fa rotar l'objecte i amb dos dits es fa zoom. Aquesta adaptació només tenia efecte en la finestra de visualització, no permetent l'ús dels menus de MS Windows. Això ha demostrat ser un inconvenient, ja que obliga a l'usuari a utilitzar el ratolí, dificultant la interacció.

Com a millores i treball futur, destacar:

- Canviar el mirall amb vidre per un que no en tingui, de manera que la imatge projectada no aparegui duplicada.
- Incloure en l'estructura de la taula uns altaveus de manera que es pugui millorar l'experiència (ja sigui amb música ambiental o amb respostes sonores per part del sistema).
- Adaptar les llibreries VTK [vtk] (implementant un nou `RenderWindowInteractor`) perquè puguin rebre els events TUIO.

Bibliografia i referències

- [AiPd] "Apple iPod ", <http://www.apple.com/ipod/specs.html> (darrer accés: Setembre 2009)
- [AiPn] "Apple iPhone", <http://www.apple.com/iphone/specs.html> (darrer accés: Setembre 2009)
- [BuM86] W. Buxton and B. Myers, "A study in two-handed input," *SIGCHI Bull.*, vol. 17, 1986, pp. 321-326.
- [Bux09] W. Buxton, 2009. Multi-Touch Systems that I Have Known and Loved. <http://www.billbuxton.com/multitouchOverview.html>. (Darrera accés: Setembre 2009)
- [CCV] Community Core Vision - Resum - NUI Group Community Code, <http://nuicode.com/projects/tbeta>. (Darrer accés: Setembre 2009)
- [Chu] B. Chun, "flosc: Flash Open Sound Control « benchun.net.", <http://www.benchun.net/flosc/>. (Darrer accés: Setembre 2009).
- [CrB08] A. Crevoisier and C. Bornand, "Transforming daily life objects into tactile interfaces," in *EuroSSC '08: Proceedings of the 3rd European Conference on Smart Sensing and Context*, 2008, pp. 1-13.
- [DiL01] P. Dietz and D. Leigh, "DiamondTouch: a multi-user touch technology," *Proceedings of the 14th annual ACM symposium on User interface software and technology*, Orlando, Florida: ACM, 2001, pp. 219-226.
- [EnvL] Narrow Dimmable LED Ribbon Flex by the 5-meter reel (High Density InfraRed 850 nm) http://www.environmentallights.com/products/12705/Dimmable_LED_Ribbon_5m_reel_IR850_390. (Darrer accés: Setembre 2009)
- [Han05] J.Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," *Proceedings of the 18th annual ACM symposium on User interface software and technology*, Seattle, WA, USA: ACM, 2005, pp. 115-118.
- [HIB07] S. Hodges, S. Izadi, A. Butler, A. Rrustemi and B. Buxton, "ThinSight: Versatile multi-touch sensing for thin form-factor displays," in *UIST '07: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, 2007, pp. 259-268.
- [HP150] "HP-150 - Wikipedia, the free encyclopedia.", <http://en.wikipedia.org/wiki/HP-150> (darrer accés: Setembre 2009)

- [JGA05] S. Jordà, M. Kaltenbrunner, G. Geiger and R. Bencina, "The reacTable*," in *Proceedings of the International Computer Music Conference (ICMC 2005)*, 2005.
- [JGM08] Carme Julià, Dèbora Gil, Enric Martí, "Entorn bàsic amb interfície MFC i Status Bar (PFC)", *PRACTICA GRAFICS PER COMPUTADOR 2*, Gener 2009.
- [KaB99] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. of the 2nd IEEE and ACM Int'l Workshop on Augmented Reality (IWAR)*, 1999, pp. 85–94.
- [KBB05] M. Kaltenbrunner, T. Bovermann, R. Bencina, and E. Costanza, "TUIO - A Protocol for Table Based Tangible User Interfaces," *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005)*, Vannes, France: 2005.
- [KGH85] M.W. Krueger, T. Gionfriddo, and K. Hinrichsen, "VIDEOPLACE - an artificial reality," *SIGCHI Bull.*, vol. 16, 1985, pp. 35-40.
- [LBS85] S. Lee, W. Buxton and K. C. Smith, "A multi-touch three dimensional touch-sensitive tablet," in *CHI '85: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1985, pp. 21-25.
- [LeC] LED Calculator - Current limiting resistor calculator for LED arrays
<http://ledcalculator.net/default.aspx?values=9,1.6,20,40,0>. (Darrer accés: Setembre 2009)
- [Lee07] Johnny Chung Lee, "Wii Projects", <http://johnnylee.net/projects/wii/> (Darrer accés: Setembre 2009).
- [Lem] "JazzMutant - Lemur, multitouch modular controller for sequencers, synthesizers, virtual instruments, vj and light.",
http://www.jazzmutant.com/lemur_features_hard.php (Darrer accés: Setembre 2009)
- [MaR97] N. Matsushita and J. Rekimoto, "HoloWall: designing a finger, hand, body, and object sensitive wall," *Proceedings of the 10th annual ACM symposium on User interface software and technology*, Banff, Alberta, Canada: ACM, 1997, pp. 209-210.
- [Mca86] McAviney, P. The Sensor Frame - A Gesture-Based Device for the Manipulation of Graphic Objects. Carnegie- Mellon University, 1986.
- [MPX] "X.Org Wiki - Development/Documentation/MPX",
<http://www.x.org/wiki/Development/Documentation/MPX>. (Darrer accés: Setembre 2009).

- [MSDN] "Windows Touch",
<http://msdn.microsoft.com/en-us/library/dd562197%28VS.85%29.aspx>. (Darrer
 accès: Setembre 2009).
- [Nui06] "NUI Group - Natural User Interface Group", <http://www.nuigroup.com>. (Darrer
 accès: Setembre 2009)
- [OpT] "OpenTouch", <http://code.google.com/p/opentouch/>. (Darrer accès: Setembre 2009)
- [PeM02] Perski, H., and Morag, M. Dual function input device and method. *U.S. Patent No. 6762752*. 2002.
- [Rek02] J. Rekimoto, "SmartSkin: an infrastructure for freehand manipulation on interactive surfaces," *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, Minneapolis, Minnesota, USA: ACM, 2002, pp. 113-120.
- [Ret14] "reactIVision 1.4", <http://reactivision.sourceforge.net/>. (Darrer accès: Setembre 2009).
- [San08] "MTmini - DIY Multitouch Mini Pad | SethSandler",
<http://sethsandler.com/multitouch/mtmini/>. (Darrer accès: Setembre 2009)
- [She05] B. Sherwood, "Status of PLATO IV," *SIGCUE Outlook*, vol. 6, pp. 3-6, 1972.
- [Sur07] "Microsoft Surface",
<http://www.microsoft.com/surface/Pages/Product/WhatIs.aspx> (darrer accès: Agost 2009)
- [ToL] "TouchLib", <http://nuigroup.com/touchlib/>. (Darrer accès: Setembre 2009)
- [vtk] "VTK – The Visualization Toolkit", <http://www.vtk.org/>. (Darrer accès: Setembre 2009).
- [Wel91] P. Wellner, "The DigitalDesk calculator: tangible manipulation on a desk top display," *Proceedings of the 4th annual ACM symposium on User interface software and technology*, Hilton Head, South Carolina, United States: ACM, 1991, pp. 27-33.
- [WFM03] M. Wright, A. Freed, and A. Momeni, "OpenSound Control: state of the art 2003," *Proceedings of the 2003 conference on New interfaces for musical expression*, Montreal, Quebec, Canada: National University of Singapore, 2003, pp. 153-160.

[Wii] “Nintendo Wii” –

http://www.nintendo.es/NOE/es_ES/systems/datos_tcnicos_1072.html (Darrer accès: Setmebre 2009)

[Wil04] A.D. Wilson, “TouchLight: an imaging touch screen and display for gesture-based interaction,” *Proceedings of the 6th international conference on Multimodal interfaces*, State College, PA, USA: ACM, 2004, pp. 69-76.

[Wil05] A.D. Wilson, “PlayAnywhere: a compact interactive tabletop projection-vision system,” *Proceedings of the 18th annual ACM symposium on User interface software and technology*, Seattle, WA, USA: ACM, 2005, pp. 83-92.

Annex A – Components

En aquest annex es presenten les característiques més importants dels components utilitzats en la construcció de la taula multitàctil definitiva:

| Component | Projector Epson EMP-600 |
|----------------------|---------------------------------------|
| Mida | 34.8 cm x 10.41 cm x 27.43 cm |
| Resolució | 480.000 píxels (800x600)x3. SVGA. 4:3 |
| Brillantor | 1700 ANSI lumens |
| Contrast | 400:1 |
| Profunditat de color | Color verdader. 24 bits. 16M Colors |

Taula A.1: Característiques del projector

| Component | Càmera Sony PS3Eye |
|-------------------------|---|
| Resolució | 640x480 píxels @ 60 Hz 320x240 píxels @ 120 Hz |
| Connexió | USB 2.0. Connector tipus A |
| Angle de visió | De 56° a 75° (variable)s |
| Profunditat de color | 32 bits |
| Format de la imatge | Comprimida en JPEG i sense compressió |
| Sensor i interfície USB | OmniVision OV538 |

Taula A.2: Característiques de la càmera

| Component | Cinta de leds de Environmental Lights |
|-----------------------|---------------------------------------|
| Alimentació | 12 V |
| Potència | 48 W |
| Angle de visió | 120° |
| Amplada de la cinta | 8 mm |
| Alçada (cinta i leds) | 2 mm |
| Distància entre leds | 13 mm |
| Color | Infrarojos de 850 nm. |

Taula A.3: Característiques de la cinta de leds

Annex B – Llibreries reactTable

En aquest annex s'exposarà com utilitzar la llibreria del reactTable per tal de poder rebre i utilitzar els events TUIO. Per fer-ho, s'explicarà com implementar una aplicació que imprimeixi la posició dels dits per consola en C++.

El primer que cal, és descarregar-se la llibreria del reactTable des de [Ret14]. Dins de l'arxiu comprimit es troben dues carpetes amb el codi necessari per rebre les dades via xarxa i per implementar els events:

- Carpeta oscpack
- Carpeta TUIO

Un cop copiades aquestes dues carpetes allà on es tingui el projecte o arxiu main.c, es pot començar a implementar l'aplicació.

Per tal d'implementar els events, caldrà crear una classe, en l'exemple classe *Exemple*, que derivi de *TuioListener*. Per tal de poder-ho fer, caldrà incloure l'arxiu *TuioListener.h* i *TuioClient.h*. Aquesta derivació obliga a definir i implementar les funcions corresponents als events de fer contacte, deixar de fer contacte i moviment, tant pels dits com pels objectes (ja que aquesta llibreria permeten treballar amb objectes). A més, caldrà definir que utilitzem el *namespace* TUIO. Així, l'arxiu *Exemple.h* queda:

```
#ifndef INCLUDED_EXAMPLE_H
#define INCLUDED_EXAMPLE_H

#include "TuioListener.h"
#include "TuioClient.h"
#include <math.h>

using namespace TUIO;

class Exemple : public TuioListener {
public:
    void addTuioObject(TuioObject *tobj);
    void updateTuioObject(TuioObject *tobj);
    void removeTuioObject(TuioObject *tobj);

    void addTuioCursor(TuioCursor *tcur);
    void updateTuioCursor(TuioCursor *tcur);
    void removeTuioCursor(TuioCursor *tcur);

    void refresh(TuioTime frameTime);
};

#endif /* INCLUDED_EXAMPLE_H */
```

La funció `refresh()` normalment no s'utilitza, i per tant no es comentarà.

Com es pot veure en el codi anterior, les funcions reben un punter de tipus *TuioCursor* o *TuioObject*. Aquestes dues classes estàn definides en els arxius *TuioCursor.h* i *TuioObject.h* dins de la carpeta TUIO, respectivament. Totes dues deriven de la classe *TuioContainer* que deriva de *TuioPoint* (definides en els arxius *.h* amb el mateix nom dins de la carpeta TUIO).

Començant per *TuioPoint*, aquesta classe representa un punt dins de la superfície de contacte. Les variables membre més important són *xpos* i *ypos* (coordenada x i y). Aquí cal remarcar que aquestes coordenades estàn normalitzades, de manera que van de 0 a 1. Així, l'aplicació que les utilitzi ha de saber la resolució de la pantalla per convertir-les a coordenades pantalla (simplement multiplicant-les per l'amplada i l'alçada d'aquesta, respectivament). A més, aquesta classe implementa funcions per calcular la distància i l'angle respecte un altre *TuioPoint*.

La classe *TuioContainer*, derivada de *TuioPoint*, afegeix una capa d'abstracció. Així, la variable membre més important és *path* (encara que aquesta classe també té informació sobre velocitat de moviment, però no es tindrà en compte en aquest exemple). La variable *path* és una llista de *TuioPoints*. Aquesta llista representa el conjunt de punts pels que ha passat el dit o l'objecte, en el seu moviment per damunt de la superfície. Les principals utilitzats d'aquesta variable és poder recollir gestos o poder mirar si dos punts s'acosten o s'allunyen, per exemple.

Finalment, les classe *TuioCursor* i *TuioObject*, permeten distingir si el que ha fet contacte és un dit o un objecte (aquesta distinció la fa el *tracker* i envia missatges *2DCur* o *2DObj*. Veure capítol 2.2.3). En aquest exemple s'utilitzaran només els events dels dits (els dels objectes són quasi iguals, amb l'única diferència que tenen informació de l'angle amb que estàn col·locats). Per tal de distingir els diferents dits que poden estar en contacte damunt de la superfície, aquesta classe té la variable membre *cursor_id* que conté el valor numèric que els identifica.

Fins ara s'ha vist la informació que ens ofereix la llibreria per tal d'implementar les funcions dels events dels dits. Per una banda implementa les classes necessaries per representar els dits, el punt de conacte d'aquests i el camí que han anat seguint. Per l'altra, ens ofereix una classe amb les funcions que seran cridades en cada event. Així, el codi de les funcions membre de la classe *Exemple* és:

```
//Funcio que es crida quan apareix un contacte nou
void Exemple::addTuioCursor(TuioCursor *tcur) {
    std::cout << "add cur " << tcur->getCursorID()
    << tcur->getX() << " " << tcur->getY()
    << std::endl;
}
```



```

//Funcio que es crida quan es mou un dit
void Exemple::updateTuioCursor(TuioCursor *tcur) {
    std::cout << "set cur " << tcur->getCursorID()
    << tcur->getX() << " " << tcur->getY()
    << std::endl;
}
//Funcio que es crida quan desapareix un contacte (s'aixeca el dit)
void Exemple::removeTuioCursor(TuioCursor *tcur) {
    std::cout << "del cur " << tcur->getCursorID()
    << std::endl;
}

```

Per altra banda, cal que el programa rebi les dades del *tracker* via xarxa. Per fer-ho, la llibreria inclou la classe *TuioClient*, definida en l'arxiu *TuioClient.h* i implementada en l'arxiu *TuioClient.cpp*, tots dos dins de la carpeta *TUIO*. Aquesta classe fa ús del codi que conté la carpeta *oscpack*, on hi ha el codi necessari per interpretar les dades OSC via xarxa.

Utilitzant el patró de disseny d'observador, la classe *TuioClient* fa d'observable i la classe *TuioListener* d'observador. D'aquesta manera, quan la classe *TuioClient* rep una missatge TUIO per xarxa, crida la funció corresponent a l'event de la classe *TuioListener*. Així, la funció *main* quedaria:

```

int main(int argc, char* argv[]) {
    if( argc >= 2 && strcmp( argv[1], "-h") == 0 ){
        std::cout << "usage: Client [port]\n";
        return 0;
    }

    int port = 3333;
    if( argc >= 2 ) port = atoi( argv[1] );

    Exemple exe;
    TuioClient client(port);
    client.addTuioListener(&exe);
    client.connect(true);

    return 0;
}

```

Actualment, les interfícies multitàctils estan guanyant popularitat, tant entre els usuaris com entre els desenvolupadors i fabricants. En aquest projecte s'estudien les interfícies multitàctils basades en visió per computador: el seu funcionament, la seva contrucció i la implementació del software que permet la interacció amb més d'un dit simultàniament. En la memòria es presenta aquest estudi, com s'ha construït una taula i els resultats que s'han obtingut amb ella.

Actualmente, la interficies multitactiles estan ganando popularidad, tanto entre los usuarios como entre los desarrolladores i fabricantes. En este proyecto se estudian la interficies multactiles basadas en visión por computador: su funcionamiento, su contrucción y la implementación del software que permite la interacción con mas de un dedo simultanimanete. En la memoria se presenta este estudi, como se ha construido una mesa i los resultados obtenidos con esta.

Currently, multitouch interfaces are gaining popularity, both among users and between the developers and manufacturers. This project focuses on multitouch interfaces based on computer vision: their performance, construction and deployment of software that allows interaction with more than one finger simultaneously. In this study report is presented how a table has been constructed and the results that have been obtained with it.